# COMPUTER GENERATION OF TELEVISION IMAGES
# REAL-TIME SIMULATION OF LUNAR LANDSCAPE DISPLAYS (U)

By

Joseph C. McMenamin

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Marshall Space Flight Center
Alabama 35812

Contract NAS8-21367

May 1969


By

PENNSYLVANIA RESEARCH ASSOCIATES INC.
101 North 33 Street
Philadelphia, Pennsylvania 19104

# COMPUTER GENERATION OF TELEVISION IMAGES: REAL-TIME SIMULATION OF LUNAR LANDSCAPE DISPLAYS (U)

By

Joseph C. McMenamin

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Marshall Space Flight Center
Alabama 35812

Contract NAS8-21367

May 1969

By

PENNSYLVANIA RESEARCH ASSOCIATES INC.
101 North 33 Street
Philadelphia, Pennsylvania 19104

## ABSTRACT

This is a technical report on the results of a one-year contract to study the application of techniques for real-time computer generation of television images to the visual display of a Lunar Surface Roving Vehicle (LSRV) Simulator at National Aeronautics and Space Administration, Marshall Space Flight Center (NASA, MSFC). Many of the techniques herein have been known to simulation system designers for some time. This report outlines a system that combines known techniques of data representation and hybrid (analog-digital) computation with some very powerful and novel methods of electronic image synthesis. Although the techniques are specifically applied to the LSRV simulator, they are also useful in many other types of real-time display generation. The Appendix to this report includes the results of an effort to simulate in non-real-time the type of out-the-window displays which might be produced by the techniques described in this report.

## ACKNOWLEDGEMENT

The present report incorporates the results of work performed by several staff members of Pennsylvania Research Associates Inc. (PRA); the named author is the originator of most of the written material herein presented. Mr. Lawrence Lieberman is the originator of the Lunar Simulation Computer Program described in the Appendix.

TABLE OF CONTENTS

TABLE OF CONTENTS continued

## LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

# COMPUTER GENERATION OF TELEVISION IMAGES: REAL-TIME SIMULATION OF LUNAR LANDSCAPE DISPLAYS (U)

## I. INTRODUCTION

An important part of many real-time vehicle simulators is the visual display which provides the operator with a simulated out-the-window view of the vehicle's environment. Indeed, the visual display is often the most difficult and costly portion of a simulator. This is particularly true for the Lunar Surface Roving Vehicle (LSRV) Simulator being developed by NASA at Marshall Space Flight Center (MSFC). The present report develops the requirements for a digitally controlled out-the-window display and means of implementing it by use of electronic, as opposed to physical, models.

In the past, the basis of such visual displays has been a TV camera-model system in which a TV camera, mounted on a movable gantry, views a painted, photographic, and/or relief model of the vehicle's environment. NASA's present Lunar Surface Roving Vehicle (LSRV) Simulator, a modified SMK-23, is of this type. Specifically, the SMK-23 uses a 3" image orthicon camera and a rubber relief map of a portion of the lunar surface. The camera is located over the model by servos which are controlled by analog computers. The input to the computers is a set of terrain sensors which detect the height and slope of the terrain at the position of the vehicle.

The limitations and drawbacks of simulators based on a physical model are immediately apparent. First of all, they are very cumbersome and require a considerable amount of space. For example, at a scale of 150 to 1 a square mile of terrain requires a 35 foot square model. Next, the speed and acceleration which can be simulated are limited by the speed and strength of the servos which position the TV camera. Also, there are many limitations in the TV system, for example, resolution, depth of field, adequate lighting, noise, etc. Finally, they are susceptible to the failure of their numerous mechanical parts. Because of these limitations, methods were sought to replace the physical model with a mathematical model in a computer.

Early attempts at computer image generation in real-time were, however, limited to producing pictures which were composites of simple geometric shapes described by either vectors or elementary functions. Although it has also been possible with a computer to enhance single frames obtained by photographic processes, this is presently not practical in real-time.

Recently, Pennsylvania Research Associates Inc. (PRA), under contract to the Naval Training Device Center (NTDC), has developed methods of producing real-time, computer-generated TV images which have grey scale and texture rather than solid geometric shapes. These images are produced by special-purpose hybrid* computing equipment under the control of a general-purpose digital computer. At the present time, NTDC is constructing a digitally controlled Radar Landmass Simulator using PRA techniques. Technical Reports on PRA's development of radar simulation techniques are listed in the bibliography.

Although a visual simulator creates somewhat different problems, many of the basic techniques of radar display synthesis are applicable to it. Therefore, MSFC contracted with PRA to develop techniques for real-time computer generation of a TV display for NASA's Lunar Surface Roving Vehicle (LSRV) Simulator. Specifically, the requirement is to:

(1) devise methods of using hybrid computation of signals to complement the images now produced by the present LSRV Simulator using a TV model, and

(2) devise methods of computing the complete television image in real time to obviate the TV model for the LSRV Simulator.

---

* Throughout this report "hybrid" computation refers to equipment which exhibits both analog (or continuous) and digital (or discrete) computing properties.

The work of the present contract is an initial investigation, emphasizing those methods of (1) which would also be applicable to (2) above. This report contains the results of the study and an initial recommendation of a system for producing a computer-generated, real-time TV display for the LSRV Simulator.

The remainder of this report is divided into three main sections. The first of these is a statement of the data and computational requirements of a LSRV Simulator. The next section explains the techniques which are recommended to compute the required TV signals for the simulator in real time. The last section contains specific methods of complementing the present TV display of the LSRV Simulator with computer image generation techniques. The Appendix to the report contains the results of a recent attempt by PRA to simulate in non-real-time the type of TV picture which would be produced by the methods recommended in this report.

## II. SIMULATOR CONCEPTS

### A. Simulator Requirements

The visual simulator techniques described in this report are specifically tailored for use in a Lunar Surface Roving Vehicle (LSRV) Simulator. They are also adapted to simulating a vehicle which "flies" over the lunar surface. The major differences between the visual display of a surface vehicle and a "flying" vehicle are the area displayed at a given time and the size of the smallest object on the ground which is resolvable. It will be seen that the use of the proposed techniques allows enough flexibility that the same simulator hardware might be used to simulate both types of vehicles or possibly even a combination roving-flying vehicle. The requirements of both types of simulator are listed below.

Before going into the technical description of lunar simulator techniques, it is appropriate to avoid confusion by defining some of the terms which will be used in this report. The term "display" refers to the simulated scene which the "driver" of the vehicle looks at. This display is a composite of "display elements" which are calculated by the simulator. The nature of the display elements depends on the type of display hardware which is used. For example, if a television CRT is used, the display elements are the TV sweep waveforms, 500-1,000 per frame.

When calculating picture elements it is often convenient to take a projection on some reference plane of the scene being viewed. This plane is called the view-plane. When discussing the lunar landscape the following terms are convenient. "Lunoid" is used to describe the sphere having the same radius as the mean radius of the moon. All calculations of the lunar landscape are made relative to this sphere. The slow variations of the landscape are referred to as the "terrain". Terrain includes large mountain ranges, maria, and slow undulations. Lunar "features" are the smaller variations of the landscape. Craters, small mountains, rills,

scarps, are all classed as features. Non-single-value features such as rocks are classed as "polyhedral features". A special sub-class of the features is designated as "culture". This class includes man-made objects such as landing vehicles, equipment stockpiles, etc. A special characteristic of "culture" is its ability to be moved, or added to or deleted from the display during the course of a mission. Culture can also have a specular or directional light reflection characteristic rather than the almost diffuse reflection of the terrain. "Landscape", of course, refers to the composite of lunoid, terrain, features, and culture. The total area of the lunar surface for which data is stored is referred to as the "problem area".

The display for the LSRV is to be an "out-the-window" display covering $50^{\circ}$ horizontal by $30^{\circ}$ vertical field of view. When the vehicle is on level ground this allows the driver to see no closer than about twenty feet, with the horizon appearing to be a little less than 3 statute miles distant. High mountains,however,can be seen from much farther. For example, a mountain that is a quarter of a mile high can be seen by an observer on the surface of the lunoid who is over 30 statute miles away. If desirable, the display could also include the more important stars and the earth as aids to astronomical navigation. It is important that shadows cast by the terrain and features as well as by the vehicle itself be included. The very small shadows cast by irregularities in the lunar features are taken care of by introducing a "texture" parameter for each of the various features.

Since the display hardware is usually fixed in position relative to the driver, it becomes necessary to change the display as the vehicle changes its attitude. For example, when the vehicle pitches upward, the scene must move down at the proper rate to give the illusion that the vehicle is moving. The display can compensate for roll, pitch, and yaw without making a completely new calculation of all of the display elements, since the relative positions of objects do not change when the attitude is changed. Changes in height, latitude, or longitude, however, require that

all picture elements be recomputed, since in this case the apparent relative positions on the view-plane of the objects viewed will change. A flying vehicle is capable of the same maneuvers as the roving vehicle, although at faster rates. Something should be mentioned here about the rates of the various vehicle motions. For a roving vehicle, the maximum lunar speed is about 16 ft/sec. and turning rates can be as high as $20^{\circ}$/sec. The high maneuverability of the flying vehicle creates speeds an order of magnitude higher and turning rates on the order of $5^{\circ}$/sec. The problems associated with these rates of motion are more apparent below where changes in the display associated with these motions are examined.

B. Data Required to Produce the Display

From a theoretical point of view, the display is a projection of the light reflectivity of the lunar landscape upon the view-plane. For convenience in computation this process is looked at in the opposite sense, namely, each display element is projected down to the lunar landscape and the light reflectivity from that area is used for the intensity of the display element. The projection is illustrated in Fig. 1(a).

This projection procedure is probably the most complicated job of the simulator. It must account not only for changes in vehicle orientation, but also for the varying height of the landscape. The varying heights cause parts of the distant landscape to be blocked from view or "occluded" by nearer portions of the landscape. This problem is shown explicitly in Fig. 1(b) where only part of the landscape is visible (indicated by a heavy line).

The light reflectivity can be either calculated from the combined landscape data or else separately (and perhaps differently) calculated for terrain, features, etc. and the results appropriately superimposed on the display. The most feasible method of accomplishing this is discussed below within the context of the total simulator system.

The light reflectivity is computed from knowledge of remote landscape slope and relative positions of the driver and sun. The addition of shadows to the reflectivity data is done in two parts, remote shadow and local shadow. Fig. 2 should help to distinguish between the two types of shadow. The local shadow is the shadow on the side of a section of the landscape away from the sun and the remote shadow is the shadow cast by an object onto a point of the landscape which would not normally be in shadow if it were isolated from the rest of the landscape.

In a more mathematical sense, a point on the landscape is said to be in local shadow when the scalar product of the unit normal vector to the landscape at the point with a unit vector to the "sun" from the point is greater than zero. A point on the landscape is in remote shadow if

- it is not in local shadow and
- a line from the point to the sun intersects the landscape at least once between the point and the sun.

The reasons for this distinction will become clear when the overall system is considered.

Therefore, it is seen that to produce the display, it is necessary to know

- the orientation of the vehicle
- the projection of the display elements onto the terrain
- the reflectivity from each part of the terrain.

The orientation of the vehicle is computed from the slope of the local landscape* below the vehicle. The projection of the display elements is computed from the vehicle orientation and the height and slope data of the remote landscape. Various methods of implementing this calculation are examined in Section III.C. below.

*It is convenient here to distinguish between "local landscape" which influences the orientation of the vehicle and "remote landscape" which is visible to the driver.

The light reflectivity and local shadow are calculated from the slope of the remote landscape and the positions of the sun and the vehicle. Remote shadow is calculated for the entire problem area before each "mission".

Thus the data needed on hand to compute the display is

- vehicle position and local slope
- remote height and slope data
- sun position
- remote shadow data

Although this seems obvious, it should be kept in mind throughout the discussion so that no details are overlooked. It is now appropriate to examine the changes which occur in the display data as the image changes.

C. Changes in Display Data

As the vehicle moves, and new terrain becomes visible, it is of course necessary to replace some of the data used to produce the previous frame. Since the landscape data for the entire problem area represented is probably too large to be kept in the main computer memory at all times, only the data needed for the calculation of the frame at hand can be extracted from the complete landscape data bank and put into the computer memory. Obviously, much of the data in the main computer memory from a previous frame can be used with some new data, to compute the present frame. An efficient means of replacing this relatively small amount of data, while keeping the data which doesn't change, is necessary since the frame rate (30 frames/second) is too fast to allow a total replacement from the data bank of all the data for each frame.

The simplest type of vehicle motion is moving forward on level terrain. In this case, the distant landscape appears not to change while

the foreground moves by quickly. Hence, the local height and slope data change but the area for which the remote height and slope data are required changes only very slowly, even when the vehicle moves at maximum speed. This can be seen in Fig. 3. Notice also that a smaller area is used as the vehicle progresses. This is an ideal situation since the same data is used repeatedly to calculate successive frames of the display.

Another good case is when the vehicle pitches. If it pitches upward, less of the landscape is seen and no new data must be supplied to calculate the display. A downward pitch, on the other hand, introduces a need for new data very close to the vehicle. This area (shown dotted in Fig. 3) is very small, and might be carried along at all times even when it is not used.

Rolling changes only the orientation of the display relative to the display screen and is easily accounted for by a simple coordinate transformation. Fig. 4 clearly shows this. Again if only a little extra data is covered along with the necessary data, no new data will have to be supplied to generate the display when the vehicle rolls.

Unfortunately, when the vehicle turns, the data can change entirely in very little time. Fig. 3 shows how this happens. Besides changing quickly, turning is also different in that the background changes most rapidly and the foreground relatively slowly. This rapid change of a large volume of data can be handled by methods described later in this report. The basic idea behind all of the methods is the fact that the rapidly changing background appears blurred to the driver and therefore need not be calculated as precisely as the more slowly changing foreground. It is also likely that the attention of the driver will be focused on the foreground during a turn and the background will go unnoticed. Regardless of whether the data in the main computer memory must be changed, the problem still exists of processing the data in the memory into a display at a fast enough rate to avoid flicker and jerkiness of the display image. Ideally, the display should be completely regenerated with new data (updated) at a rate of sixty frames per second. Because of the large

number of calculations which have to be made it might be difficult or impossible to obtain this rate. If this is the case, the frame refresh rate could be reduced to thirty per second without introducing flicker and the display data update rate could be reduced to as low as ten frames per second before the jerkiness becomes objectionable. Since the update and refresh rates at which the display can operate depend, of course, on the details of the system driving the display, further discussion is left to Section III on overall system configuration.

## D. Calculations and Mathematical Models

It has been shown that it is necessary to know the landscape height and slope data to compute the display. The way in which the various parts of the landscape are represented is described below. Here it is assumed that the required height and slope data is available in the computer and the calculation of light reflectivity, shadow, occultation, etc. is covered. The way in which these calculations are implemented is discussed below in the section on overall system configuration.

The first quantity to be computed is the illumination of the terrain by the sun. Lambert's Law states that the illumination of a point on a surface is equal to the intensity of the sunlight times the scalar product of a unit normal vector at the point with a unit normal vector pointing toward the sun from the point. Since knowledge of the terrain slope data is assumed this calculation can easily be made. Fig. 5 shows a diagram illustrating Lambert's Law. On the assumption that the lunar landscape is a perfectly diffuse reflector of light, the light reflectivity from the lunar landscape is simply proportional to the illumination, regardless of the orientation of the driver relative to the landscape. If, however, the slight specularity of the lunar surface is taken into account, an additional factor multiplied by the illumination must be used. This factor is some function of the angle between the line of sight to the point on the landscape and the normal vector to the

landscape, and also the angle between a unit vector to the sun and the normal to the landscape. Fig. 6 illustrates this calculation. Since the slope of the landscape and direction of the sun is assumed known at every point, it is possible to compute the normal vector to the landscape, the vector to the sun, and the vector to the observer. It is assumed from here on, however, that no correction is necessary for the slight specularity of the landscape. If at any time it is decided to include such a correction, it is easy to add.

The calculation of local shadow is done in exactly the same way. However, if the scalar product of the normal to the landscape with the vector to the sun is found to be negative, then that portion of the landscape is known to be in shadow and the intensity is recorded as zero. It should be noted that all of the calculations of light reflectivity in shadow made up to this point, can be calculated from only a knowledge of terrain slope at the point of interest. Remote shadow, on the other hand, requires a knowledge of terrain at points other than the point of interest. Because of this, it seems wise to compute all of the remote shadows for a given sun position before the start of a mission. The reason for this will become clearer when the overall system configuration is described.

Fig. 7 illustrates how the remote shadow is calculated. A profile of the lunar landscape is taken along a section of the landscape in a direction parallel to a ray from the sun. Starting at the edge of the section nearest to the sun, the scalar product of the normal to the terrain and the vector to the sun is computed. At the point where this quantity first becomes less than zero, the height of the terrain is noted. While continuing along the section of landscape, the height of the landscape at every point is compared to the computed height of a ray from the sun to the point at which the shadow started. When the difference between these heights is equal to zero, the intersection of the ray from the sun with terrain has been found. This point is the end of the remote shadow cast by the landscape. By repeating this procedure for many cross-sections to the landscape the entire area of remote shadow can be computed. The boundaries of this area are stored in the memory of the computer. When the final projection of the light reflectivity of the landscape onto the viewplane is made, any region which is inside of the stored shadow boundary is treated as having zero light reflectivity.

The problem of occultation is handled in much the same way as remote shadow. The main difference is that the sections of landscape are made parallel to a line of sight of the vehicle driver, rather than the direction of a ray from the sun. In this case, while moving along a landscape profile, the scalar product of the normal to the landscape with a vector toward the driver's eye is taken. When this product becomes less than zero, the height of the point is recorded and the intersection of the line of sight from the driver's eye through that point, to the next point on the landscape is computed in the same manner as for remote shadow above.. This procedure is illustrated in Fig. 8.

Since not all of the small variations in the landscape can be stored in a computer memory, it is necessary to add realism to the display by computing some form of texture to be added to the projection of the landscape. This texture can be generated in a pseudo random way which is reproduced only while it is visible on the screen. Since variations in texture are extremely small, the fact that they are different when viewed at some other time, will be insignificant and unnoticeable. The reader should also notice that it is necessary to add this texture only in the foreground of the landscape, since at large distances detail this fine is not visible. Another method of adding texture is to divide the problem area into regions and use the same texture pattern within each region. This randomly generated texture could then be stored in a read-only memory and used repeatedly.

## E. Display Hardware

At present raster displays are much superior in writing speed to any other type of display. The usual raster formats are: TV raster based on a cartesian coordinate system; PPI raster (such as radar) which uses polar coordinates; and sonar raster using a spiral sweep. The major advantage to raster format is that the beam position is a function of time relative to some synchronizing signals so that explicit beam positioning data need not be generated. Of these formats, TV raster is the most convenient from the standpoint of computation.

Displays capable of displaying one million ($10^6$) resolvable points in 1/60 second are easily within the present TV technology. Digitally driven displays which actually display one point at a time are also capable of producing an image with $10^6$ points. This type of display, however, takes several seconds to produce a complete frame -- entirely too slow for generation of a moving display. A third type of display creates an image with vectors or multiple Lissajous figures.. A typical vector display can generate one to two thousand vectors in 1/60 sec. Such a picture produced by a thousand or so vectors lacks much of the detail of a TV raster display. The other disadvantage of the latter two display formats is that they require the computer to generate explicit beam positioning data as well as intensities.

TV raster displays are also superior from the standpoint of available technology and compatibility. Many types of standard equipments for mixing, switching, and storing TV raster signals are readily available. In a sense, use of TV raster displays is backed by many years of technological development of television techniques. As well as being standard, many of the TV equipments mentioned are presently in use in existing simulators and display systems.

A specific reason for the use of TV raster format is synchronization. The steady and sequential flow of data in a TV raster signal allows many operations to be made at the same in synchronism and combined together in available video mixing networks.

The specific type of TV raster display to be used depends on the physical setup of the simulator. The displays can be divided into two main types -- projected displays and so-called "infinity optics" displays. The first type (of which the Eidophor display is an example) simply projects the raster on a screen placed perhaps twenty feet in front of the viewer. This is far enough way that the driver can focus his eyes at infinity when viewing the display.

The "infinity optics" displays(such as the several types manufactured by Farrand Optical Company) are placed close to the driver and rely on optics to create the illusion of distance and depth. Although the appropriate TV display must be chosen to be compatible with the physical layout of the simulator, TV displays all require essentially the same input signals: intensity and deflection in raster format.

The only disadvantage of the two displays described above is that some type of display sweep memory is needed if the display need not be refreshed for every display frame. Although new types of displays which don't require this are being developed, none are feasible at present.

## III. OVERALL SYSTEM CONFIGURATION

### A. Data Representation in Storage

There are several different ways in which the various height and slope data can be represented and stored. The more important methods are listed below with their advantages and limitations.

#### Two-dimensional Polynomial Representation

With this method, the surface to be represented is divided into "regions" within which the variations in the height are neither too large nor too numerous. A least squares fit of a polynomial (probably of fourth to sixth order) is made to the given terrain data (if actual empirical data is used), and the coefficients of the polynomial are stored for each region. Under contract to NTDC, PRA has made an extensive study of this technique and found that the most feasible way to store the polynomial is in terms of the coefficients of the Lagrange polynomial representation. For this purpose, the region sizes would probably be chosen as a few hundred meters square. It might also be advisable to store the terrain in two levels of representation, one having regions several times larger than the other. This would reduce the amount of data required to produce the profiles of distant parts of the terrain. In addition, height profiles of the higher mountains might be stored since at distances much over a mile only such height profiles would be visible. The Lagrange polynomial itself is of the form:

$$h(x,y) = \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{p=0}^{2} \sum_{q=0}^{2} g_{ip}\left(\frac{X-X_k}{d}\right) g_{jq}\left(\frac{Y-Y_l}{d}\right) f_{i+k,\ j+l,\ p,q}$$

where k and l define the region, i and j define the corners of the regions (see Fig. 9), d is the region size, and p and q determine the order of f.

The terms $g_{ip}$ and $g_{jq}$ are of the form

$$g_{00}(Z) = (1 - Z)^3 \ (1 + 3Z + 6Z^2)$$

$$g_{01}(Z) = (1 - Z)^3 \ (1 + 3Z)$$

$$g_{02}(Z) = \tfrac{1}{2}(1 - Z)^3 Z^2$$

$$g_{10}(Z) = Z^3 (6Z^2 - 15Z + 10)$$

$$g_{11}(Z) = Z^3 (1 - Z) (3Z - 4)$$

$$g_{12}(Z) = \tfrac{1}{2}Z^3 (1 - Z)^2$$

and

$$f_{i+k,\ j+l,\ p,\ q} \Longleftrightarrow \left. \frac{\partial^{p+q} n}{\partial u^p \partial v^q} \right|_{\substack{u = i+k \\ v = j+l}}$$

where $f_{i+k,\ j+l,\ p,\ q}$ are the stored coefficients.

The derivatives are with respect to u and v rather than x and y so that within a region the effective values of x and y are normalized to travel from zero to one. This eliminates the necessity of using geographic coordinates within a region. In this "modified" form which uses derivatives at corner points, the slope of the terrain located at the boundary between regions is continuous across the boundary. Plots of the $g_{rs}$ are shown in Fig. 10. These basis fuunctions are common to all regions.

It is interesting to note that $g_{0j}(Z) = |g_{1j}(1 - Z)|$. This symmetry feature will become particularly useful later.

As a sidelight, the indices p and q can reach the general values of m and n respectively. For the region size described below it was decided that $m = n = 2$ would produce sufficient accuracy for the application at hand. Hence the designation "2-2 version of the modified Lagrange polynomial". With this the case then, the polynomial itself contains 36 terms; each term has associated with it a derivative, the order of which is the highest combination of p and q. Hence the cross derivative

$$\frac{\partial^4 h}{\partial u^2 \partial v^2}$$

Now instead of storing individual point heights across a region, what is stored in the polynomial representation are the 36 derivatives associated with each region. As a convention, 9 are stored at each region corner and are arranged such that all of these constants are valid ones for the four adjacent regions associated with that corner. Fig. 11 shows pictorially the placement of polynomial coefficients at region corners. When applied to the appropriate products of the basis functions and then summed, the 36 coefficients (stored derivatives) completely characterize the terrain with the region.

Consider briefly the thirty-six terms of the polynomial itself. They can be arranged into four groups of nine terms each; each group corresponds to a region corner. Consider a sweep (straight line) passing through a region corner. At any given boundary the only terms in the polynomial which contribute to its value at that point are those located at the two corners describing the boundary. Hence if a crossing is made at $x = 0$, $y = .5$ the only $g_{rs}$ products which are non-zero are those which are of the form

$$g_{0p}\, g_{0q} \quad \text{and} \quad g_{0p}\, g_{1q}$$

Notice that the first subscript of each $g_{rs}$ corresponds to the appropriate corner designation. In other words $g_{0p}\, g_{0q}$ is associated with the corner $(0,0)$ and $g_{0p}\, g_{1q}$ is associated with the corner $(0,1)$. See Fig. 9. This is to say that at a boundary effectively only half of the terms of the polynomial are contributing elements, a characteristic which proves to be very important in the implementation of the HPG discussed below.

## 2. Special Function Representation

Another type of representation is the special function. This is a function which is specifically formulated to represent a certain type of feature. It is particularly useful for features which occur frequently and which are very similar, such as craters, rills, etc., but it is also applicable to any single-valued features which are required to look smooth and are too small to be included in the terrain representation. The special function is similar to the two dimensional polynomial representation except that each feature is treated separately and the parameters are

quantities like position, height, diameters, type, etc, rather than polynomial coefficients. In addition, the ground coordinates of the vertices of the rectangle bounding the feature are needed. A special purpose hybrid "feature generator" could then be made to convert these digital parameters into analog height and light reflectivity "profiles" to be combined into the video output signal. A classification or representation scheme for craters using the technique described above has been developed by PRA in an effort coordinated with this project. This method of crater representation is described below. Several examples of the type of images which can be generated by a computer using this method are shown in the Appendix.

It was decided that the majority of craters can be classed into one of three general types. The cross-section along a diameter of each of these types is shown in Fig. 12. The mathematical formula describing each type is given below.

The shape of a type one crater of Fig. 12(a) is approximated by a circular parabaloid. In terms of the radial distance $\rho$ from the center of the crater the height is given as:

$$h = \frac{\rho^2}{\beta} - \alpha$$

where: $\alpha$ = depth of the crater
$\beta$ = shape of crater.

To put this formula in terms of parameters which are more easily measured directly from a map (namely depth, radius, and center coordinate) the above formula is converted by using:

$$\rho^2 = (X - X_0)^2 + (Y - Y_0)^2$$

where $(X_0, Y_0)$ is the center coordinate. Since the crater radius $r_1$ (radius at which the height is zero) is:

$$r_1 = \sqrt{\alpha\beta}$$

this yields:

$$h = \left[ \frac{(X-X_0)^2 + (Y-Y_0)^2}{r_1^2} \right] \alpha$$

within the region where $h < 0$.

As can be seen from Fig. 12(b) the second type of crater is like a type one crater within a radius $r$, and has a rim of height $\delta$ inside an annular sector $r_1 \leq \rho < r_2$. Mathematically, the equation of the rim can be written as:

$$\left[ 1 - \left( \frac{2\rho - r_2 - r_1}{r_2 - r_1} \right)^2 \right] \delta \quad .$$

The complete formula for a type two crater is:

$$h = \begin{cases} \left[ \dfrac{(X-X_0)^2 + (Y-Y_0)^2}{r_1^2} - 1 \right] \alpha \quad , \quad 0 \leq (X-X_0)^2 + (Y-Y_0)^2 < r_1^2 \\ \left[ 1 - \left( \dfrac{2\sqrt{(X-X_0)^2 + (Y-Y_0)^2} - r_2 - r_1}{r_2 - r_1} \right)^2 \right] \delta \quad , \quad r_1^2 \leq (X-X_0)^2 + (Y-Y_0)^2 < \end{cases}$$

The type three crater illustrated in Fig. 12(c) is simply a type two crater with an extended flat bottom. The equation describing a type three crater is:

$$h = \begin{cases} -\alpha \quad , \quad 0 \leq (X-X_0)^2 + (Y-Y_0)^2 < r_1^2 \\ \left[ \left( \dfrac{\sqrt{(X-X_0)^2 + (Y-Y_0)^2} - r_1}{r_2 - r_1} \right)^2 - 1 \right] \alpha \quad , \quad r_1^2 \leq (X-X_0)^2 + (Y-Y_0)^2 < r_2^2 \\ \left[ 1 - \left( \dfrac{2\sqrt{(X-X_0)^2 + (Y-Y_0)^2} - r_3 - r_2}{r_3 - r_2} \right)^2 \right] \delta \quad , \quad r_2^2 \leq (X-X_0)^2 + (Y-Y_0)^2 < r_3^2 \end{cases}$$

### 3. Polyhedron Representation

Another useful type of representation is approximation by a polyhedron. The main advantage of this type of representation is that it can be used to represent a surface which is not single valued. This is particularly useful for representing rocks and man-made features which are not visually single valued. The major limitation is the amount of data required to represent a large or detailed feature. The vertices of the polyhedron and the unit normal to each face must be stored. This is sufficient data to calculate the height and reflectivity profiles.

Note that the use of polyhedron representations should be restricted to convex figures to avoid the "hidden line problem". It is also convenient to assume that the objects represented as polyhedra are almost all small enough relative to the terrain that they can be considered to be either entirely visible or invisible. This assumption greatly simplifies the implementation of the display calculations. Large or complex objects represented by polyhedra (e.g., a spacecraft) require new techniques not developed at the time of this report. Fig. 13 shows a typical rock represented as a polyhedron.

## B. Organization and Operation of the System

Presented above is the list of data necessary to produce a vehicle visual display, the calculations which are made with this data, and the way in which the landscape data is represented in storage. How the necessary calculations are made on the data in real time by a digitally-oriented hybrid computer is now examined.

A block diagram of the overall simulator system is shown in Fig. 14. The Digital Processing Unit (DPU) is the main control and storage unit of the simulator. It supplies the necessary data to the Terrain Generator (TG), Feature Generator (FG), Shadow Controller (SC), Texture Generator (TXG), and Polyhedron Generator (PG) in synchronism with each other. The outputs of these various hybrid generators are combined in the Mixing Unit (MU) to produce a video signal which can be applied to a TV display tube. Before going into the details of the various system components it is instructive to briefly explain the operation of this simulator.

At the beginning of a simulated mission, the DPU is given the initial position of the vehicle, the position of the sun, and the direction in which the vehicle is pointing. The DPU then calculates and stores the remote shadow boundaries (previously explained) and the polyhedron reflectivities (explained below). The simulator is then ready to proceed with the mission. While the mission is underway, the DPU calculates the vehicle attitude from the local height and slope data at the given initial position. Based on this information the DPU next calculates the region of visibility for which remote height and slope data is needed. This data is transferred into the hybrid generators which produce height and reflectivity analog signals for the landscape along the projection of a display element (this projection is also a line-of-sight of the driver). These signals are time-varying analog voltages whose amplitudes are proportional to height or reflectivity at a point on the landscape and whose "time" is proportional to the "ground range" of the point along the line-of-sight.

These synchronized time-varying voltages are combined together in the Mixing Unit to produce a composite signal which is the total light reflectivity as a function of position on the display plane. This gives one complete set of intensities for a TV raster sweep. The process is repeated for each sweep until the entire frame is completed. Then the data to be sent to the hybrid generators is revised and new display element projections are calculated.

## C. Implementation of Calculations

### 1. Landscape Profiles

It is mentioned above (in Section I.D) that the way to avoid the problems of landscape occultation is to calculate along landscape sections which are taken parallel to the driver's line-of-sight (hereafter such sections are referred to as "landscape profiles"). Assuming a TV raster format, this means taking the sweep lines in a vertical direction so that the projection of a display element (raster scan) is just one of these landscape profiles. In this way all the calcu-

tions are made along each profile, resulting in an intensity vs. time signal which can be applied directly to the TV display.

When the vehicle rolls, the projection of a display element is no longer a true landscape profile. The most straightforward way to avoid this problem is to rotate the deflection yoke on the display tube to maintain display elements which are true landscape profiles. Since the rate of roll is small (a few degrees per second) and the total roll angle never exceeds $40^{\circ}$ this is quite feasible. Alternatively, the terrain function generator could be modified to accept such "rolled" profiles. In any case, this is not a serious problem and it is assumed therefore, from hereon, that the roll is compensated for and that it is not necessary to consider it further. A geared-down stepping motor, directly driven by pulses from the digital computer, is as feasible for this purpose as an analog servo.

## 2. Organization of the Digital Processing Unit (DPU)

A block diagram of the Digital Processing Unit is shown in Fig. 15. The DPU is the main control and storage unit for the simulator. The Central Processing Unit (CPU) in the DPU is a standard third-generation digital computer which is strongly I/O oriented. Besides controlling the real-time generation of the display the CPU is used to do the remote shadow and polyhedron reflectivity calculations before the start of the mission. These data are stored in the respective shadow and polyhedron data banks. (See dotted lines in Fig. 15.)

During the real-time operation of the simulator the CPU must perform the following task. At the start of each frame the CPU calculates which portion of the problem area is visible to the driver. It then tells the I/O processors what data they must transfer from their respective data banks into their storage buffers to compute the desired landscape profiles. This transfer into buffers is needed since the data banks must be very large and therefore probably too slow to transfer the necessary data directly into the hybrid generations in real time. Initial

considerations suggest that drum storage is the most feasible device for the data banks. The CPU also generates control signals which are sent to the hybrid generators and Mixing Unit to keep the various hybrid calculations in synchronism.

It is the job of the I/O processors to edit the data transferred from the data banks and to order it so that it is ready to be transferred to the proper hybrid generator at the instant it is needed. The exact allocation of tasks between CPU and I/O processors depends, of course, on the specific hardware which is used. For example, if a very powerful CPU is used, the job of editing and ordering might be done by the CPU with a corresponding reduction in the sophistication of the I/O processor.

To allow for features which can change during the course of the mission, the CPU merely replaces the data in the data bank describing that feature with the changed data. This technique might be useful in adding vehicle tracks to the display by storing a small patch of remote shadow at each point along the path of the vehicle.

The data outputs from the DPU which are shown in Figs. 14 and 15 are as follows:

1. Digital Terrain Polynomial Coefficients

2. Digital Feature Parameters

3. Digital Shadow Boundary Data

4. Digital Polyhedron Vertex and Reflectivity Data

5, 6. Digital Data and Control Signals for Synchronism

7. Digital Vehicle Roll Signals

## 3. Implementation of the Terrain Generator

Assuming that the proper sets of polynomial coefficients have been assembled by the DPU, the way these coefficients are converted into analog height and reflectivity profiles, and ultimately into the resultant video signal, can be examined. Generally, the TG will compute a height and reflectivity profile by evaluating the height and slope of the terrain from the Lagrange polynomials along a projected display element. Because of the similarity of the height and reflectivit calculations, the height profile generation is described in detail and the reflectivity profile technique modeled after it.

It is most convenient to start the height profile calculations from the point directly below the driver's eye, even though the first part of this profile will not be used. The angle $\theta_n$ in Fig. 16 has previously been calculated, and the origin of the driver's eye is known.

This origin will be within some region (K,L), at a "local position" $(X_0,Y_0)$. The local position is always taken relative to the size of the region i.e., normalized to the region size. With this information, as well as the computed angle $\theta_n$ (shown in Fig. 16) of the $n^{th}$ display element projection, the complete equation of the line can be written as a parametric equation of the dummy variable $t$. Namely:

$$X = X_0 + t\cos\theta \qquad 0 < X < 1$$

$$Y = Y_0 + t\sin\theta \qquad 0 < Y < 1$$

In any other region a similar parametric equation can also be written. A typical sweep through a region is illustrated in Fig. 16.

Consider this region (K,L) with its thirty-six associated coefficients as shown in Fig. 17. As long as the sweep is within the region each coefficient is contributing a value to the total

height of the surface inside. It is remembered, however, that at any boundary only the coefficients located at the corners (points) defining that boundary contribute to the value of the polynomial there. Also, because coefficients are valid in adjacent regions, the ones which are significant during a transition across a boundary can be retained for calculation in the next region.

This clean separation of contributors at a boundary is extended to the hardware itself. Examine a network which calculates the value of every term associated with one corner. For the sake of example pick the corner (0,1), of Fig. 18. The terms which must be calculated are products of two basis functions and a coefficient applicable to the region:

$$g_{0p}\, g_{1q}\, f_{0,1,p,q}$$

The $f_{0,1,p,q}$ is a constant which is output from the digital computer; the $g_{rs}$ are polynomials in $x$ or $y$ which are produced by analog function generators.

Fig. 19 shows the physical layout for calculating the values of terms associated with the example corner. Notice that only three distinct types of function generator are used; these are

$$g_0\ ,\ g_1\ ,\ g_2$$

Consider that a sweep takes time $t \cos \theta$ to cross the two vertical boundaries of the sample region; it should be noted that this time is constant from vertical boundary to vertical boundary. Transforming along the $x$ axis the time of the $x$ sweep region crossing is $t \cos \theta$; the period of the $y$ sweep is $t \sin \theta$. The inputs to the function generators are therefore ramps having slopes proportional

to cos $\theta$ and sin $\theta$, where $\theta$ is the azimuth of the terrain profile being reconstructed. The signal for region boundary crossing (new coefficients needed) is one ramp's value being equal to the maximum (a limit stop). Now in order to produce $g_{00}(X)$, one feeds a $g_0$ generator with $X(t)$. Furthermore to give $g_{10}(X)$, one supplies $X_1(t) = 1 - X(t)$ to a $g_0$ generator. This makes use of the symmetry property described earlier.

Hence the entire network consists of twelve function generators -- three of which are distinct. Three of the function generators ($g_0$, $g_1$, $g_2$) are fed $X(t)$; three are fed $Y(t)$; three are fed $1 - X(t)$; and three are fed $1 - Y(t)$. The appropriate outputs are multiplied together and then multiplied in a D/A multipler by the coefficients $f_{0,1,p,q}$ and all 36 terms summed to give height.

The function generators need not be duplicated for each corner. Their outputs, however, must be combined appropriately for the other corners to produce the correct combinations of $g_{ip}\, g_{jq}$.

Now the system consists of four such multiplying matrices all of whose outputs are summed to produce the height profile for the sweep. There are 12 function generators, 36 analog multipliers, and 36 D/A multipliers. The inputs are sawtooths and digital numbers; the output is the height profile.

Consider carefully the boundary that the sweep crosses as it leaves the region (See Fig. 18). The values of x and y may or may not have reached a maximum; in the case of the present example X is at its highest value, 1. The value then of $X(t)$ must fall to zero and new coefficients must be brought in for the new region. At present there are two sets of coefficients which could be used, namely those associated with the boundary. However they are loaded

into the wrong D/A multipliers in the hardware so they must be transferred. These two abrupt changes produce undesirable transients in the output which cannot be tolerated.

Now keep the coefficients on the boundary, change the ones not on the boundary, and cause the sweep to go backwards. In other words instead of a sawtooth wave for X(t), Y(t) there is a triangular wave which effectively can be thought of as creating a sweep which never leaves a given region. When a boundary is reached, however, new coefficients are brought in on the opposite corners and the sweep is reversed. The same desired output of height profile is achieved and advantages have been gained -- 1) the inputs to the function generators do not have undesirable transients, and 2) the output from the TFG is continuous.

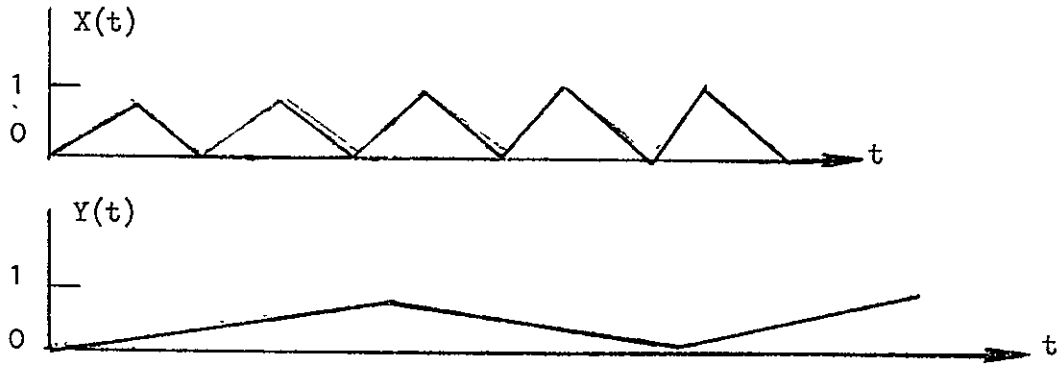
Now continue this thinking with the aid of a drawing (See Fig. 20). Consider a region (marked with *) in which a sweep starts. Label each corner with a colored dot[1]. As the sweep crosses a boundary label the new region with the colors not associated with the boundary. Hence pt. 1 would be colored green and pt. 2 would be colored red. Every time a boundary is crossed two new corners must be colored and also two new sets of coefficients must be brought in. Continue this process for a few sweeps.

Very rapidly one sees an interesting trend (See Fig. 21). In some horizontal lines there are only red and black dots. In alternate horizontal lines there are only green and yellow dots. Similarly there are alternate red/green and black/yellow dots in the vertical lines. This property has conveniently produced a generalized ordering scheme for retrieving the appropriate coefficients from storage. This will be returned to below; now continue the analysis of the colored dot structure.

---

[1]For each of reproduction, the colors in Figs. 21 and 22 are represented by letters: B = black; R = red; Y = yellow; G = green. It is suggested that the reader fill in the diagrams with actual colors to aid in following the description.

A very significant fact is that one can determine what color the new dots should be when a boundary is crossed. As example, consider the waveforms of X(t) and Y(t)




together with the sweep shown in Fig. 21. Notice that the point where X(t) = 1 corresponds to coloring in a yellow and black set of corners; X(t) = 0 corresponds to a red and green set, similarly in Y. The following table summarizes the possibilities.

| | |
|---|---|
| X = 0 | color next dots red and green |
| X = 1 | color next dots yellow and black |
| Y = 0 | color next dots red and black |
| Y = 1 | color next dots green and yellow |

Note the reference to "red and green set" which implies partitioning the data into an easily retrievable form. This is exactly the case. In storage four arrays (corresponding to red, black, green, and yellow) are formed. All that is needed then is a pointer to indicate the position in the array. The next "red dot" will be either 1 in X or 1 in Y away from the pointer.

Now return to the hardware. If each corner in the multiplying matrix is defined by a color, then it is noticed that only red data will be fed into the red corner, only green data into the green corner, etc. Therefore, only a switch and three registers need be added to each D/A multiplier to completely automate the system. Fig. 22 shows this addition. The center register is a holding register. The other two load the next values of data corresponding to a boundary reached first in X or first in Y. The switch is controlled by the inputs (X(t), Y(t)), and it selects which register is to be transferred into the hold register. This selection is made according to the same criterion as coloring dots before. The table below again summarizes the possibilities.

| Transition | Corner Selection | Switch |
|---|---|---|
| X = 0 | transfer red and green | X First |
| X = 1 | transfer yellow and black | X First |
| Y = 0 | transfer red and black | Y First |
| Y = 1 | transfer green and yellow | Y First |

In addition to transferring, the above table of transitions generates interrupts to the computer which tell it to add integer 1 (in X or Y) to the appropriate colored array points, which in turn will load the appropriate XFIRST and YFIRST registers. Hence the system is completely self-structured and requires no previous calculations of region crossings or sweep path.

As an incidental point, the azimuth of the sweep referenced to the region boundaries is $\theta$ and is the only calculation made, in addition to locating the first region (setting up pointers), at the beginning of each sweep. Also the generation of the triangular waves is done in an accumulator with a D/A converter with a smoother on the output. In this way the sweeps are in sync with the digital computer.

Using this same technique the normal vector to the terrain at each point within a region can be computed. The formula for the normal vector n is:

$$\hat{n} = \left(\hat{k} - \frac{\partial h}{\partial x}\,\hat{i} - \frac{\partial h}{\partial y}\,\hat{j}\right) \frac{1}{\left[\left(\frac{\partial h}{\partial x}\right)^2 + \left(\frac{\partial h}{\partial y}\right)^2 + 1\right]^{\frac{1}{2}}}$$

Remember that $h = h(x,y)$ = height of terrain at the local position $(x,y)$ within a region.

$\hat{n}$ can be obtained directly in terms of polynomial coefficients by using time derivatives of Lagrange polynomials since:

$$\frac{\partial h}{\partial x} = h'_x = \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{p=0}^{2} \sum_{q=0}^{2} g'_{ip}(x)\, g_{jq}(y)\, f_{i+k,\, j+l,\, p,\, q}$$

$$\frac{\partial h}{\partial y} = h'_y = \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{p=0}^{2} \sum_{q=0}^{2} g_{ip}(x)\, g'_{jq}(y)\, f_{i+k,\, j+l,\, p,\, q}$$

Where:

$$g'_{00}(z) = (1-z)^2 \quad (6 + 18\,z + 6\,z^2)$$

$$g'_{01}(z) = (1-z)^2 \quad (1 + 8\,z + 3\,z^2)$$

$$g'_{02}(z) = \tfrac{1}{2}\,(1-z)^2\, z\, (z-2)$$

$$g'_{10}(z) = 30\, z^2\, (z^2 - 2\,z + 1)$$

$$g'_{11}(z) = z^2\, (-15\, z^2 + 28z - 12)$$

$$g'_{12}(z) = \tfrac{1}{2}\, z^2\, (1-z)\, (3-z) \quad .$$

and the $f_{i+k,\, j+l,\, p,\, q}$ are the polynomial coefficients.

If the sun has azimuth angle $\alpha$ (from x-axis) and elevation angle $\epsilon$ (from horizontal), then the vector to the sun is, from Fig. 23,

$$\hat{S} = \text{Sin } \epsilon \; \hat{k} + \cos \epsilon \; (\cos \alpha \; \hat{i} + \sin \alpha \; \hat{j}).$$

Then, assuming that the lunar landscape is a diffuse reflector, the reflectivity $r$ can be generated from polynomial coefficients and sun position data as:

$$r(x,y) = (\sin \epsilon - h'_x \cos \epsilon \cos \alpha - h'_y \cos \epsilon \sin \alpha) \; [1 + (h'_x)^2 + (h'_y)^2]^{-\frac{1}{2}}$$

where $h_x$ and $h_y$ have been defined above as components of terrain slope.

The entire calculation of $r(x,y)$ is done in a hybrid array with the same method as used for height profiles. Thus the output of the TG will be two time varying analog voltages representing height and reflectivity respectively. Fig. 24 shows these height and reflectivity profiles as functions of time (or ground range) as they might come out of the TG together. They are continuous across region boundaries, which are thus invisible, as a result of the polynomial representation used.

Incidentally, if more than one level of terrain representation is used (as discussed in Section II.A) the Terrain Generator must be slightly modified. The Modified Terrain Generator (MTG) is essentially two single TG's whose outputs are switched to the Mixing Unit depending on the ground range of the terrain profile. In this way it is necessary to load the high resolution TG only coefficients describing the high resolution regions close to the vehicle, thus reducing the total number of coefficients which must be transferred for a given sweep. sweep.

### 4. Implementation of the Feature Generator

The method for handling features represented by special functions (e.g., craters) is similar to the method used to handle

the terrain. The Feature Generator (FG) is composed of a number of sub-generators each of which is capable of generating a specific type of feature. When the sweep is within the region bounding a certain feature, the digital parameters describing that feature are loaded into the appropriate sub-generator within the FG and the x and y analog sweep voltages applied to that sub-generator. The output of the sub-generator will be two time varying voltages; one representing height vs. ground range and the other representing slope vs. ground range. It is thus necessary to have a separate type of sub-generator for each type of feature and some way of combining the outputs of the individual sub-generators.

All of the height output voltages are added together into a single feature height vs. ground range analog voltage. The slope signals are added vectorally to form a vector slope signal which can be multiplied (scalar product) by a vector to the sun. This yields a feature reflectivity vs. ground range analog voltage. These two voltages are the output of the FG to the Mixing Unit.

There are two ways to organize the FG. The difference between them depends on the allocation of computation between the special-purpose FG and the DPU. To reduce the burden on the DPU the features are stored and loaded in groups of features inside of regions. These regions are conveniently chosen to be the same as the regions used in the terrain representation. Using this approach the special purpose equipment must have enough sub-generators to take care of all of the features within a region or else it must have some digital computing and memory capability for loading the correct set of parameters into the correct sub-generator at the right time. If, on the other hand, the amount of special-purpose equipment is to be reduced at the expense of more calculation by the DPU, only enough sub-generators need be used to account for any nesting (superposition) of features while the DPU calculates when and where to load all of the parameters.

One simplification which can be made in the FG is to assume that no two features lie on top of each other. In this case the

reflectivity can be computed separately for each feature instead of from the composite slope. To allow for small craters on top of other features (a frequent occurrence) more than one crater sub-generator could be employed and the outputs of these sub-generators appropriately switched to get a composite output.

## 5. Implementation of the Shadow Controller

It is the job of the Shadow Controller (SC) to blank the reflectivity signal to insert the remote shadow. The Shadow Controller does this by calculating where the sweep crosses the stored remote shadow boundaries and alternately blanking and unblanking the sweep. There are two very different ways of implementing this -- one digital oriented and the other analog oriented.

In the digital method, the shadow boundaries, stored as a sequence of line segments in a digital memory, are loaded by the shadow data I/O processor into a hybrid computing unit which computes the intersection of the analog sweep input with the line forming the shadow boundary. This calculation is made by putting the line segment data (a point, length, and direction along the ground) into an analog "line generator". Referring to Fig. 25, the x component of the sweep is fed into the line generator and the corresponding value of y along the shadow boundary is the output. When the y component of the shadow boundary is equal to the x component of the sweep the lines are intersecting and the reflectivity signal is blanked. It is most likely that several of the line generators must be employed so that the order and speed with which the digital parameters must be loaded into line generators are less critical. In that case the outputs of all of the line generators are combined in an OR gate and then compared with the sweep voltage as explained above. A block diagram of such a controller is shown in Fig. 26.

Notice that synchronism with the terrain and feature generators is guaranteed by the fact that the same sweep voltage is used as input to both the generators and the controller.

The analog type of Shadow Controller utilizes an analog memory such as a scan converter. Here the shadow boundaries are written into the analog memory in x,y coordinates and read out along the sweep. When a shadow boundary is crossed a pulse comes out of the memory and changes the state of the flip-flop which blanks the reflectivity signal. This method is still synchronized with the other generators. However, small errors may result from poor registration of the sweep in the memory.

## 6. Implementation of the Polyhedron Generator

It is most convenient to have calculated the light reflectivity value of each face of each polyhedron and to have stored this value with the polyhedron vertices before the mission is run. This is possible assuming diffuse reflection of light from the lunar surface. Even if it is decided to include some specularity function in the terrain calculations, it could be neglected for features represented as polyhedra (hereafter referred to as "polyhedron features"). Assuming this, it is possible to compute the reflectivity vs. ground range profile to be combined with the other reflectivity profiles. This job is complicated by the fact that the intensity is a double-valued function of ground range. The way to account for this is to compute directly the projection of the polyhedron features onto the view plane. This is explained below.

The projection can be considered in two steps:

(1) a transformation of the feature vertices into a coordinate system which is fixed relative to the vehicle, and

(2) a projection from this coordinate system onto the display plane.

Note that no transformation is made to account for vehicle roll since it was previously assumed that this would be accounted for by a stepping motor rotation of the display yoke or some similar means. Fig. 27 shows how the landscape coordinate system is related to the vehicle and view plane coordinate systems.

From this figure, it can also be seen that:

- roll is a rotation ( $\Psi$ ) about U' axis
- pitch is a rotation ( $\theta$ ) about the V' axis
- heading is a rotation ( $\phi$ ) about the W axis

Notice that with this ordering heading and yaw are the same.

If each vertex of the feature is stored as a column vector

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}$$

The transformation can be written as a product of two matrices, one to account for pitch and the other to account for heading. Notice that the standard ordering is always used to avoid errors due to the non-commutation of the rotations. Thus, the transformed vector is:

$$\begin{bmatrix} u_i' \\ v_i' \\ w_i' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix}$$

If the roll is also to be accounted for in this projection it is necessary to multiply the above result by an additional roll matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Psi & \sin\Psi \\ 0 & -\sin\Psi & \cos\Psi \end{bmatrix}$$

When projected onto the display plane the vertex appears at:

$$(x_i, y_i) \quad \text{where} \quad x_i = \frac{D}{v_i' \quad u_i'} \ ,$$

$$y_i = \frac{D}{w_i' \quad u_i'}$$

These projected points determine the boundaries of the various reflectivity values on the display plane. The intersections of the reflectivity boundaries with the display sweep are computed digitally and converted into an analog signal proportional to the landscape reflectivity as a function of display plane position. This signal can then be mixed with the reflectivity vs. display plane position signal which is stored in the sweep memory. Incidentally, it should be noted that it is necessary to project only those features which will be visible to the driver. Since the polyhedron features are usually small, such as rocks, it can be assumed that these features are either entirely visible or invisible. To do this, one may define such a feature to be visible if the point on the landscape on which it rests is neither occluded nor shadowed. It is necessary to project only those polyhedra which are not too distant from the vehicle to be seen. This maximum viewing distance would have to be determined from the size of the largest polyhedron.

## 7. Implementation of the Texture Generator

An inherent property of all the above types of approximations to the landscape is that they are smooth and lacking in texture. To give realism to the display it is necessary to add some type of texture. This texture is especially important in the foreground of the picture where small details can, in practice, be seen by the driver. If such small texture is stored for the entire problem area the amount of data is enormous and the data rates impossible to provide. The solution to this problem is to store one region of a randomly generated texture pattern which is symmetric along its edges. This texture can be added directly to the reflectivity vs. ground range profile. The texture is added at this point so that it has the proper perspective when projected onto the display plane. Because of the perspective projection and the effects of occultation the fact that the texture pattern repeats itself in every region should not be noticeable. Rather than actually storing this pattern it is possible to generate the pattern in real time from some type of repeatable pseudo-random routine. This would eliminate the need for large amounts of texture storage but requires a random-function generator that operates at video frequencies.

## 8. Details of the Mixing Unit

A block diagram of the Mixing Unit (MU) is shown in Fig. 28. The Video Switch (VS) used in the MU is a standard piece of TV hardware. Referring to Fig. 29, when a signal is applied to the gate input G, the threshold detector switches the output from the normally closed input C to the normally open input S. The switch stays in this position for the duration of the signal at input G, and then returns to input C.

The first video switch (VS1) replaces the terrain reflectivity signal by the feature reflectivity signal if one is present. This corresponds to adding the features "on top of" the terrain. VS2 blanks the reflectivity signal in the presence of a remote shadow signal from the shadow controller. This is simply an addition of the shadow "reflectivity" on top of the terrain and feature signal. The texture

signal is added to the composite reflectivity signal as a high frequency perturbation. The Display Projection Processor (DPP) (explained in detail below) calculates the projection of the reflectivity profile onto the display plane, thus making it the intensity signal for the TV display tube. Since this projection is made at a constant ground range rate, the display plane rate is discontinuous*, making it necessary to store the "projected" analog signal until at least one entire raster sweep is stored in the Sweep Memory (SM). At the same time that the projected signal is read into the Sweep Memory, the third video switch (VS3) adds the signal from the polyhedron generator.

### 9. Sweep Memory

The Sweep Memory stores the display sweep signals before they are sent to the TV tube. If the memory is capable of storing the sweep signals for an entire frame, the update rate at which the frames are computed can be much less than the display frame rate. One very suitable type of memory is a dual gun cathode ray storage tube or scan converter. This accepts the signals from the computer directly and allows them to be displayed in any format (e.g., horizontal raster as opposed to vertical raster as used in the computer). An alternative to the scan converter is an A/D converter which converts the display sweeps into digital form which can be stored in a digital memory such as a magnetic disc. This can then be retrieved through a D/A converter and sent to the display tube. This has the disadvantage that the read-out raster format must be the same as that used by the computer.

A type of variable-length delay line which might also be suitable for an SM is currently under development. The SM poses an interesting technical problem due to its required speed -- equivalent to a TV sweep of perhaps 1,000 resolvable points in 3 μsec.

*Although discontinuous the signal is still sequential.

## 10. Implementation of the Display Projection Processor

As its name implies, the Display Projection Processor (DDP) forms the projection of the ground-range reflectivity profile onto the display. Fig. 30 illustrates this projection. The projection process can be looked at as the height data controlling the point to which the corresponding reflectivity data is projected. From Fig. 30 it is obvious that the elevation angle $\eta$ of the line-of-sight to a point $z$ on the terrain is given by

$$\eta(z) = \cot^{-1} \frac{z}{H - h(z)}$$

It should be noted that $\eta$ is not a single-valued function of the ground range $z$. Using the outlook given above $\eta(z)$ is the angle at which the reflectivity $r(z)$ is seen. Because of the non-single-valuedness of $\eta(z)$ more than one reflectivity can be mapped to any value of $\eta$. If the mapping is taken from the lowest value of $z$ to the highest, only the first intensity mapped to any $\eta$ is stored. This is how occultation is accounted for. This can be implemented by comparing the deriative $\eta' = \frac{\partial \eta(z)}{\partial z}$ to zero. When $\eta'$ becomes less than zero the reflectivity signal is blanked and the value of $\eta$ at that time is stored in a sample and hold amplifier. When $\eta(z)$ returns to the value stored, the reflectivity signal is unblanked. If $\eta$ is subtracted from the pitch angle $\Theta$ the resulting $\delta(z)$ is the line-of-sight angle relative to the center of the view-plane. The $y$ value of the projection on the display plane is then given by $D \tan \delta(z) = y(z)$ where $D$ is the distance to the view-plane. If the value of $y(z)$ is used to control the vertical sweep position and $r(z)$ is used to control the reflected intensity in the sweep memory, the sweep on the screen can be read in real time directly from the sweep memory. If this process is repeated for each of the sweeps making up the display, the entire picture is generated.

## IV. AUGMENTATION OF THE PRESENT SIMULATOR

As a way of demonstrating the real-time display generation capabilities of a digital computer, the below addition of vehicle shadow to the present SMK-23 LSRV Simulator is proposed. This technique for adding the shadow of the roving vehicle* to the display is very similar to the technique used to display polyhedral features in the hybrid system described above. The addition of vehicle shadow has the advantage that it can be implemented with a small digital computer and readily available video hardware.

Several other improvements to the display have also been considered, namely addition of digitally stored features (e.g., vehicle tracks) and image enhancement. Any digital image enhancement techniques are too slow to operate in real time or else are capable of only the same degree of enhancement as contrast and crispening techniques well known in television technology. The addition of digitally stored features poses a very serious problem, namely occultation of an added feature by terrain on the physical model. Since it is not possible to obtain any height data from the TV signals, such occultations cannot be computed unless the height data of the physical model is also stored digitally. Storing and processing this data would require so much additional equipment that it would be better to eliminate the physical model and calculate the entire display. Although digital terrain data is also needed for the exact computation of vehicle shadow, the approximation method explained below eliminates this need, making the addition of vehicle shadow feasible.

The coordinate systems and rotations used in the calculations are described first. Next, the assumptions used in the calculations are explained and the mathematics of the transformations presented. Finally, two methods of inserting the computer-generated shadow are suggested.

---

*It should be noted that this approach is good for any type of vehicle on or above any terrain.

## A. Coordinate Systems and Rotations

Three coordinate systems are involved in the vehicle shadow calculation. One is fixed relative to the terrain. The other two are movable with respect to the terrain but fixed relative to the vehicle. Fig. 27 shows the relation of these coordinate systems. All of the coordinate systems are right-handed rectangular cartesian coordinate systems.

### 1. Terrain Coordinate System

The coordinate axes of the system fixed relative to the terrain are defined as follows:

- The U-axis is from south to north on the terrain model.
- The V-axis is from west to east on the terrain model.
- The W-axis is from above to below (downward) on the terrain model.
- The Origin is the projection of the Center of the Vehicle onto the U-V plane.
- The Center of the Vehicle is defined to be the point about which the vehicle rotates and is where the driver's eye is located most of the time.

### 2. Vehicle Coordinate System

The coordinate axes of the system fixed relative to the vehicle are defined as follows:

- The U'-axis is from aft to forward on the vehicle.
- The V'-axis is from port to starboard on the vehicle.
- The W'-axis is from top to bottom (downward) on the vehicle.
- The Origin is the Center of the Vehicle.

The origin of the vehicle coordinate system is at (0, 0, - H) in the terrain coordinate system "H" is the center of the vehicle above the ground plane. The ground plane is assumed to be a flat plane. If the terrain features are to be taken into account, the terrain coordinate system is redefined such that the origin of the vehicle coordinate system is (A, B, - H). "(A, B)" is the point on the terrain above which is the driver's eye.

3. View-plane Coordinate System

The coordinate axes of the view-plane are fixed relative to the vehicle and are defined as follows:

- The X-axis is parallel to the V'-axis of the vehicle coordinate system (port to starboard).
- The Y-axis is parallel to the W'-axis of the vehicle coordinate system (top to bottom).
- The Origin is forward of the driver's eye at the same height in the vehicle as the driver's eye.

The origin of the view-plane coordinate system is at (D, 0, 0) in the vehicle coordinate system. "D" is the distance of the view-plane from the driver's eye*. It is the U'-coordinate of the origin of the view-plane coordinate system.

If the view-plane is not perpendicular to the fore and aft axis of the vehicle, then the vehicle coordinate system should be redefined so that the U'-axis is perpendicular to the view-plane. This will simplify the notation and calculations.

## 4. Rotations

The rotations of the vehicle are right-handed rotations about the vehicle axes. They are defined by:

$\Psi$ = roll about the U'-axis
$\theta$ = pitch about the V'-axis
$\phi$ = heading about the W-axis.

The rotations will always be taken in this order to avoid commutation errors. Note that heading and yaw are the same for this ordering.

## 5. Sun Position

The sun's position is specified by two angles, $\varepsilon$ and $\alpha$, in the terrain coordinate system. The angles are defined by:

$\varepsilon$ = elevation above the terrain
$\alpha$ = azimuth about the z-axis.

*The view-plane is conveniently chosen as any plane perpendicular to the U'-axis at a distance from the driver's eye which is less than the distance to the nearest point on the ground visible to the driver. Scaling the view-plane projection to the actual optical projection screen is accomplished by multiplying by the factor

$$\frac{D'}{D}$$

where D' is the distance to the optical projection screen.

### 6. View-plane Projection

The vehicle view-plane is considered to be parallel to the "V-W" plane.

## B. Assumptions in Calculations

### 1. Surface Representing the Terrain

Methods for storing terrain data in very compact form are discussed above. The methods consist of approximating the terrain by Lagrange polynomials or polyhedra. Although an accurate portrayal of the shadow requires such a knowledge of the terrain it is nonetheless possible to closely approximate the shadow by assuming the terrain to be flat. This approximation is unavoidable in an "add on" system for generation of the shadow since the effort required to calculate and store the terrain data is so great that it warrants the immediate adoption of a completely computerized system. That this approximation is appropriate can be seen from Fig. 31.

Notice that the angle to the shadow's edge from the view port in the vehicle depends only slightly on the height of the terrain on which the shadow falls.* It can also be seen that as the distance to the edge of the shadow increases, the difference in angle between the actual and calculated lines-of-sight decreases. Since in the present vehicles the minimum viewing range is about twenty feet**, the areas where the approximation is worst cannot be seen. This error can be calculated from the previous diagram using the law of cosines.

*Since the display is only two dimensional, the angular position is the only dimension which the observer sees.

**This assumes a flat terrain. For an upward sloping terrain the minimum range is less, for a downward sloping terrain it is greater.

Let: H = height of vehicle

h = height of view port

d = distance to shadow's edge calculated on flat terrain

z = height of shadow's edge above or below the flat terrain

Then the angular difference between actual and calculated lines of sight is:

$$\frac{z^2 \left(1 - \frac{h}{H}\right)^2}{\sqrt{\sqrt{h^2 + d^2)\left[1 - \frac{z}{H}\right)^2 d^2 + (h - z)^2\right]}}$$

Assuming typical values:

H = 10 feet
h = 5 feet
z = 2 feet

and the minimum range: d = 20 feet.

This gives 1/20 radian which is certainly negligible, for demonstration purposes*.

2. Angular Projection

This calculation is very straightforward. If a small, flat plane is assumed, the sun appears in the same position from all points on the plane. Thus, the sun position is completely specified by two angles. The angles usually given are altitude and azimuth. Notice that the azimuth angle given is measured in a clockwise direction from north.

*This approximation is still applicable to off-the-surface vehicles although it is not necessarily as good.

If the curvature of the surface must be taken into account, a more complete description must be used. This, however, should not be needed since the size of the problem area makes this correction negligible.

One further approximation made is to consider the sun as a point. This has the effect of eliminating the penumbral fringe of the shadow. This fringe is only $0.5^{\circ}$ and should be neglected. If, however, it is desired to have the penumbra, it can be calculated or the shadow could simply be "smeared" to give the same effect.

### 3. Formulation of the Characteristic Vehicle Function

It is easy to construct some function which characterizes a vehicle in some coordinate system which is fixed relative to the vehicle. In its roughest form the characteristic function is as set of points in the body coordinates which, when connected by straight lines, approximate the outline of the vehicle. The limiting case of this is, of course, a continuous function describing a closed surface (such as a set of Lagrange polynomials). This function depends only on the shape of the vehicle, not on time or rotations. Now, the problem is to transform the characteristic function to the terrain coordinates, as required for shadow calculations. The transformation is described mathematically below.

## C. Mathematical Calculation of the Shadow

### 1. Formation of the Vehicle Function

For simplicity of calculation a number of ordered points connected by line segments are used to characterize the vehicle function. This continuous string of line segments should be sufficient to describe any solid figure representation of the vehicle including non-convex and non-simply-connected ones.

To use such a string method it is necessary to also have some lines which are not needed to compute the shadow but are needed to keep the string continuous. These lines are eliminated in the final display by taking advantage of the notation method.

Thus, a typical string might look like: $a_1\ b_1\ c_1,\ 1,\ a_2\ b_2\ c_2,\ 0,\ a_3\ b_3\ c_3,\ \ldots$ indicating that the line from point 1 to point 2 is displayed but the line from point 2 to point 3 is not.

This function is given in the vehicle coordinate system and denoted $F'$. Individual points are denoted by $F'_i,\ 1 \leq i < n$.

## 2. Transformation of the Vehicle Function

The first operation is to check the orientation of the vehicle relative to the sun to see if any shadow is visible. If not, no calculations are done and the program waits for the next frame. If it is determined that some shadow is visible, then the following described calculations are initiated.

The vehicle function, $F'$, is transformed point by point to the terrain coordinates. This transformed function is denoted by $F$. This transformation involves a rotation plus a translation. In matrix notation:

$$F_i = [\phi]\ (\theta]\ [\Psi]\ F'_i + H$$

where

$$F'_i = \begin{bmatrix} U'_i \\ V'_i \\ W'_i \end{bmatrix}$$

$$H = \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix}$$

$$[\Psi] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Psi & -\sin\Psi \\ 0 & \sin\Psi & \cos\Psi \end{bmatrix}$$

$$[\theta] = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$[\phi] = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For convenience we write $[R] = [\phi]\,[\theta]\,[\Psi]$, then

$$F_i = [R]\,F_i' + H \quad .$$

The matrix transformation of $F_i$ to a set of points describing the shadow on the terrain is

$$S_i = [S]\,F_i$$

where

$$[S] = \begin{bmatrix} 1 & 0 & \cot\varepsilon & \cos\alpha \\ 0 & 1 & \cot\varepsilon & \sin\alpha \\ 0 & 0 & 0 & \end{bmatrix}$$

This is illustrated in Fig. 32.

The shadow point $S_i$ is transformed to $S_i'$ in the vehicle coordinates before it is converted to view plane coordinates. This is just the inverse of the transformation from vehicle coordinates to terrain coordinates.

$$S_i' = [R^{-1}] \quad (S_i - H) \quad .$$

Since the transformation is linear, orthogonal, and unitary: $[R^{-1}] = [R]$ and therefore $[R_{ij}^{-1}] = [R_{ji}]$. All these transformations can be combined into one which can easily be calculated on a digital computer. This transformation is applied to each point of the vehicle function F'.

From $F_i$ the shadow point $S_i$ can be found in the terrain coordinates. Since $\varepsilon$ and $\alpha$ are known, this is a simple matrix multiplication.

Let

$$F_i = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \text{and} \quad S_i = \begin{bmatrix} d \\ e \\ o \end{bmatrix} .$$

then we have:

$$r = - c \cot \varepsilon$$
$$d = a + c \cot \varepsilon \cos \alpha$$
$$e = b + c \cot \varepsilon \sin \alpha$$

The complete transformation can be written as a single matrix by writing:

$$S_i' = [R^{-1}] \, S_i - [R^{-1}] \, H$$

but $S_i = [S] \, F_i$, so by substitution:

$$S_i = [R^{-1}]\ [S]\ F_i - [R^{-1}]\ H$$

and $F_i = [R]\ F'_i + H$, so by substitution:

$$S'_i = [R^{-1}]\ [S]\ ([R]\ F'_i + H) - [R^{-1}]\ H$$

By multiplying and factoring the expression can be rearranged to read as follows:

$$S'_i = [R^{-1}]\ [S]\ [R]\ F'_i + [R^{-1}]\ ([S]\ H - H)$$

Letting: $C = [R^{-1}]\ ([S]\ H - H)$, we get

$$S'_i = [R^{-1}]\ [S]\ [R]\ F'_i + C \text{ , and :}$$

$[T] = [R^{-1}]\ [S]\ [R]$, then:

$$S'_i = [T]\ F'_i + C \ .$$

This is a very convenient form since $C$ and $[T]$ are calculated only once per frame and $[S]$ is a constant.

The projection onto the view-plane must be calculated and converted to display commands to blank the TV sweep.

The lines defined by the shadow points can be divided into four types.

(1) Entirely Visible - displayed lines that are entirely in the positive U'-axis direction.

(2) Partially Visible - displayed lines that have one endpoint with a negative U' coordinate.

(3) Not Visible - displayed lines that have no points within the view-plane at the moment.

(4) Trace Lines - lines that are never displayed but used only to store the lines defining the vehicle function.

These various types are handled as follows:

(1) Display a line between the view-plane projection of the endpoints of the line.

(2) Calculate the point where the line intersects the view-plane and connect it to the projection of the visible point.

(3) and (4) Recognize the type and then skip to the next line segment.

3. View-plane Projection

For type 1 points, by ratio and similar triangles in Fig. 33(a) :

$$\frac{X_i}{V_i} = \frac{D}{U'_i} \qquad \text{and} \qquad \frac{X_{i-1}}{V_{i-1}} = \frac{D}{U'_{i-1}}$$

so

$$X_i = \frac{D}{V'_i \; U'_i} \qquad \text{and} \qquad X_{i-1} = \frac{D}{V'_{i-1} \; U'_{i-1}}$$

If the side view is examined, the Y-axis replaces the X-axis and the W-axis replaces the V-axis in Fig. 33(a) yielding:

$$\frac{Y_i}{W'_i} = \frac{D}{U'_i} \qquad \text{and} \qquad \frac{Y_{i-1}}{W'_{i-1}} = \frac{D}{U'_{i-1}}$$

so

$$Y_i = \frac{D}{W'_i\ U'_i} \qquad Y_{i-1} = \frac{D}{W'_{i-1}\ U'_{i-1}}$$

For the projection of type 2 points, Fig.33(b) yields in the same manner as above:

$$\frac{X_i}{V'_i} = \frac{D}{U'_1} \qquad \text{so} \qquad X_i = \frac{D}{V'_i\ U'_i}$$

$$\frac{V'_{i-1} - X_{i-1}}{V'_{i-1} - V'_i} = \frac{D - U'_{i-1}}{U'_i - U'_{i-1}} \quad \text{so} \quad X_{i-1} = V'_{i-1} - V'_i)\left(\frac{D - U'_{i-1}}{U'_i - U'_{i-1}}\right)$$

If the side view is examined, the Y-axis replaces the X-axis and the W-axis replace the V-axis in the above figure and:

$$\frac{Y_i}{W'_i} = \frac{D}{U'_i} \qquad \text{so} \qquad Y_i = \frac{D}{W'_i\ U'_i}$$

$$\frac{W'_{i-1} - X_{i-1}}{W'_{i-1} - W'_i} = \frac{D - U'_{i-1}}{U'_i - U'_{i-1}} \quad \text{so} \quad X_{i-1} = W'_{i-1} - (W'_{i-1} - W'_1)\left(\frac{D - U'_{i-1}}{U'_i - U'_{i-1}}\right)$$

To determine the line type calculate if either end-point has a coordinate on the U-axis greater than D. If both do not, further calculations can be skipped.

Results of calculations for points which are used more than once can be stored for later use if some notation is used to indicate such points. For initial test runs, the computer could print out the co-ordinates of each point on the view-plane rather than produce a display.

The generation of display command depends on the type of display used. The display will have a buffer which is filled while the previous frame is being displayed.

For a non-convex vehicle, a different procedure is used. The vehicle is described as a number of convex figures each of which is separately put on the view screen.

All of the line segments of the separate convex figures are transformed to the view-plane using the above algorithm. Then the outline is calculated as follows.

The view-plane slope of each line segment is used to make this calculation. An array, headed by nodes, is made listing the lines going away from that node and the respective angles. The complement of each line is also listed. The below procedure is then carried out.

(1) Select the point with the largest y-coordinate on the screen.

(2) Select the smallest angle segment listed for that node.

(3) Store that line segment in memory.

(4) Store the complement of that line segment.

(5) Subtract the above angle from each of the angles listed for the node at the other end of the segment.

(6) Return to step (2) until the initial starting node is reached.

## D. Insertion of Vehicle Shadow into the TV Display

The shadow outline produced by the above procedure can be added using various special-purpose equipments. Two of these methods have some promise.

The first of these methods, shown in Fig. 34, blanks the TV sweep at the appropriate time to produce the correct shadow image on the display screen. The second method utilizes a scan converter to convert the digital computer output into an analog TV signal which can be mixed directly with the existing video signal, as shown in Fig. 35.

The first, or synchronized, method relies on explicit calculation of the intersection of each TV sweep with the shadow boundaries. It uses high-speed D/A converters to provide voltages for comparison with the TV deflection signals.

The second method, although simpler in concept, requires specialized TV hardware -- a scan conversion system -- to retain the shadowed area in view-plane coordinates. The scan converter serves as a buffer between the computer and the fixed-rate TV display.

## V. BIBLIOGRAPHY

1. Pennsylvania Research Associates, Inc., Investigation of Digital Techniques for Radar Landmass Simulation, Interim Rpt. Nr. 1; U. S. Naval Training Device Center, Tech. Rpt. 1025-1, Feb. 1962; AD 276 696.

2. R. L. Boyell, "The Method of Successive Grids for Reduction of Function Storage Requirements", The Computer Journal, Vol. 5, Nr. 4, Jan. 1963.

3. R. L. Boyell, "A Compression Method for Representation of Continuous Functions in a Digital Computer", presented at the Meeting of the Society for Industrial and Applied Mathematics, Stanford Research Institute, April 1963 (with H. Ruston).

4. Pennsylvania Research Associates, Inc., Investigation of Digital Techniques for Radar Landmass Simulation, Interim Rpt. Nr. 2; U. S. Naval Training Device Center, Tech. Rpt. 1025-2, Aug. 1963; AD 424 625.

5. R. L. Boyell "Hybrid Techniques for Real-Time Radar Simulation", Proceedings of the 1963 Fall Joint Computer Conference, Las Vegas, November 1963 (with H. Ruston).

6. Pennsylvania Research Associates, Inc., Investigation of Digital Techniques for Radar Landmass Simulation, Final Report; U. S. Naval Training Device Center, Tech. Rpt. 1025-3, Feb. 1964; AD 600 420.

7. Pennsylvania Research Associates, Inc., Investigation of the Compilation of Digital Maps; U. S. Naval Training Device Center, Tech. Rpt. 1025-4, Feb. 1964.

8. Pennsylvania Research Associates, Inc., Investigation of Computer Techniques for Radar Landmass Simulation (Volume 1, Analytical Investigation), U. S. Naval Training Device Center, Tech. Rpt. 1526-1, Apr. 1965.

9. R. L. Boyell, "Computer Techniques for Simulation of Air-to-Ground Radar Displays", PRA monograph, April 1967.

10. Pennsylvania Research Associates, Inc., Demonstration of Digital Radar Landmass Simulation Techniques, U. S. Naval Training Device Center Tech. Rpt. 1824-1, Oct. 1967, AD 662 407.

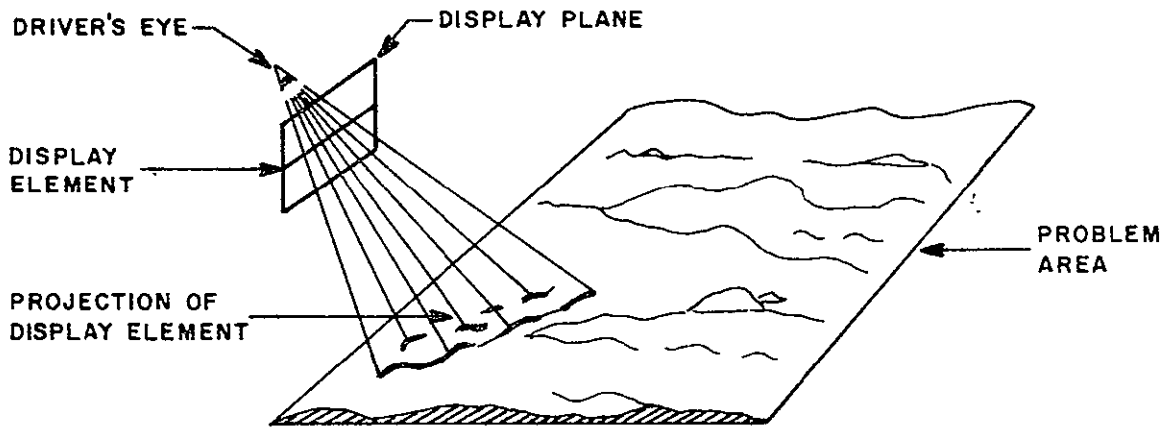
11. Pennsylvania Research Associates, Inc. Preliminary System Configuration of a Laboratory Model Radar Simulator (U), U. S. Naval Training Device Center, Rech. Rpt. 1824-2, Dec. 1967.

12. Pennsylvania Research Associates, Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 1 (U), U. S. Naval Training Device Center, Tech. Rpt. NAVTRADEVCEN 68-C-0155-1, July 1968.

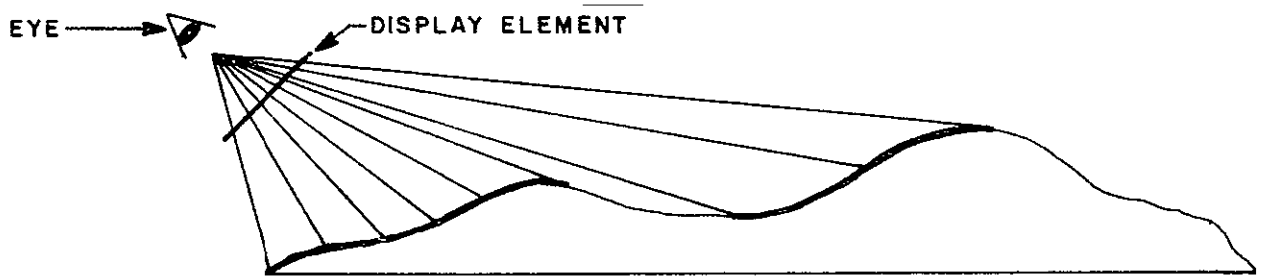
13. Pennsylvania Research Associates, Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 2 (U), U. S. Naval Training Device Center, Tech. Rpt. NAVTRADEVCEN 68-C-0155-2, Nov. 1968.

14. Pennsylvania Research Associates, Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 3 (U), U. S. Naval Training Device Center, Tech. Rpt. NAVTRADEVCEN 68-C-0155-3, Jan. 1969.

15. Pennsylvania Research Associates, Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 4 (U), U. S. Naval Training Device Center, Tech. Rpt. NAVTRADEVCEN 68-C-0155-4.




(a) PROJECTION OF ARBITRARY DISPLAY ELEMENT




(b) PROJECTION OF DISPLAY ELEMENT ALONG LANDSCAPE CROSS-SECTION

FIG. 1 - PROJECTION OF DISPLAY ELEMENT ONTO LANDSCAPE

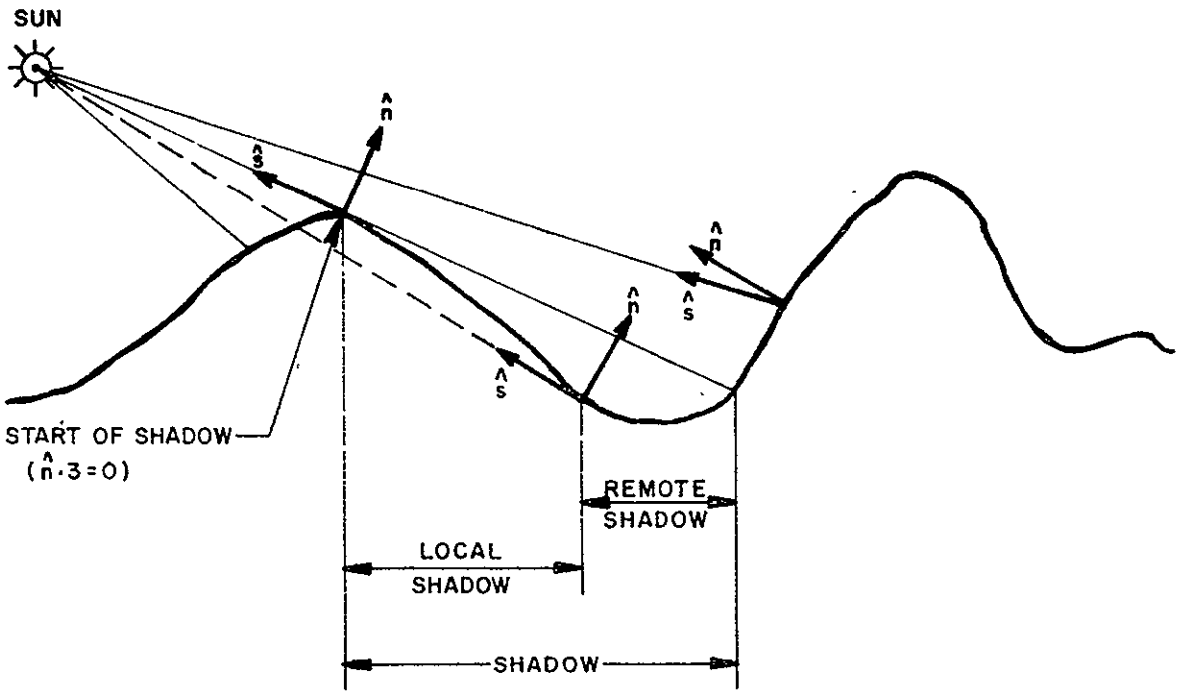



FIG. 2 – DISTINCTION BETWEEN LOCAL AND REMOTE SHADOW

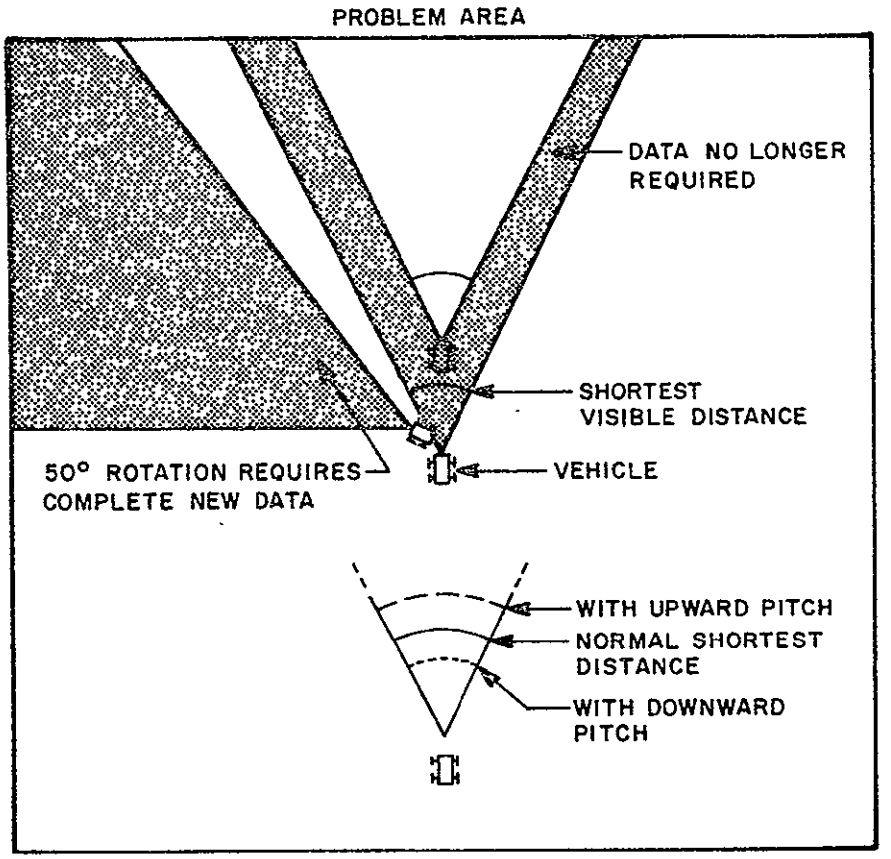



FIG. 3 – CHANGE IN DATA WITH VEHICLE MOTION

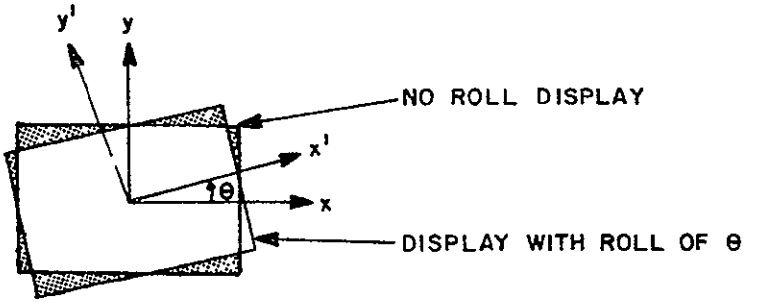



FIG. 4. – CHANGE IN DATA WITH VEHICLE ROLL

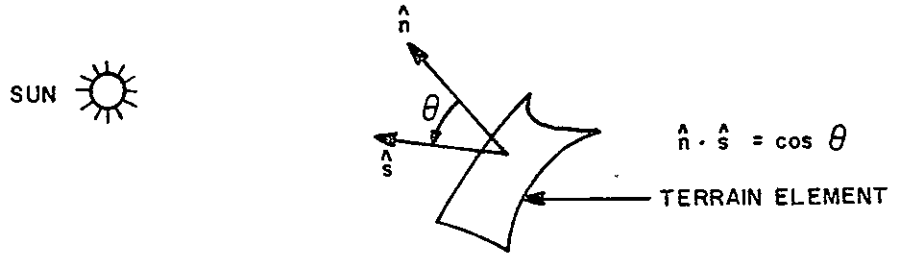



FIG. 5 - CALCULATION OF LAMBERT'S LAW OF ILLUMINATION

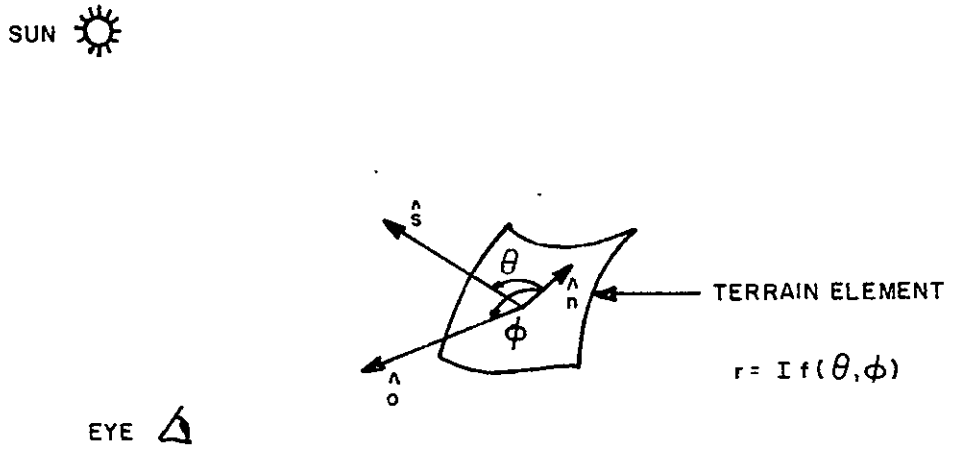



FIG. 6 - CALCULATION OF SPECULAR REFLECTION

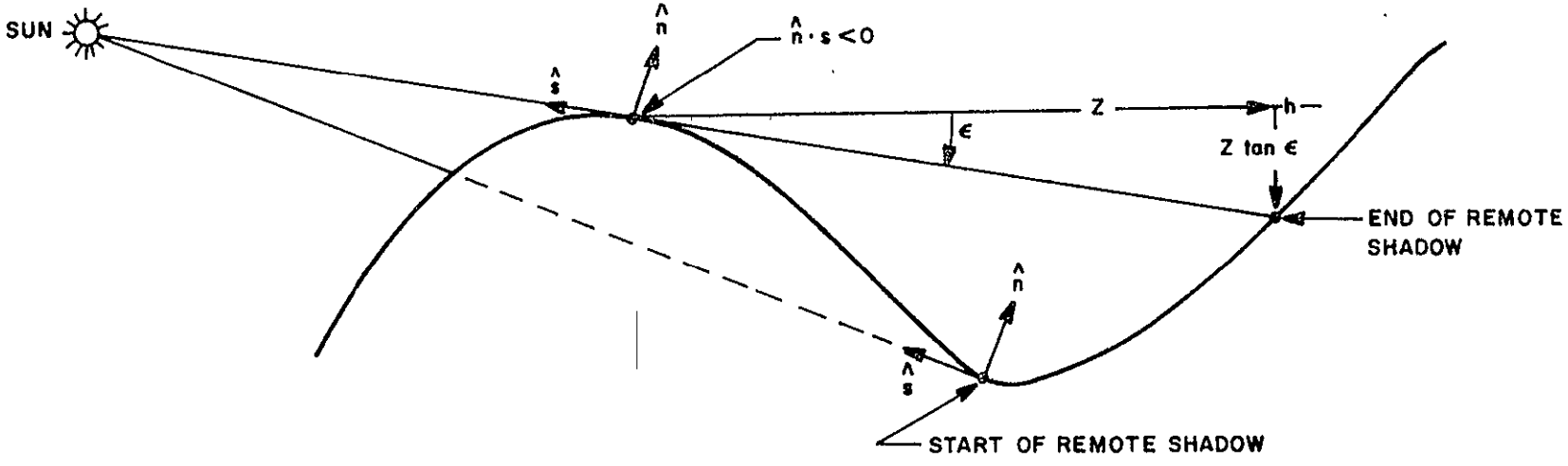



FIG. 7 – GRAPHIC ILLUSTRATION OF REMOTE SHADOW CALCULATION

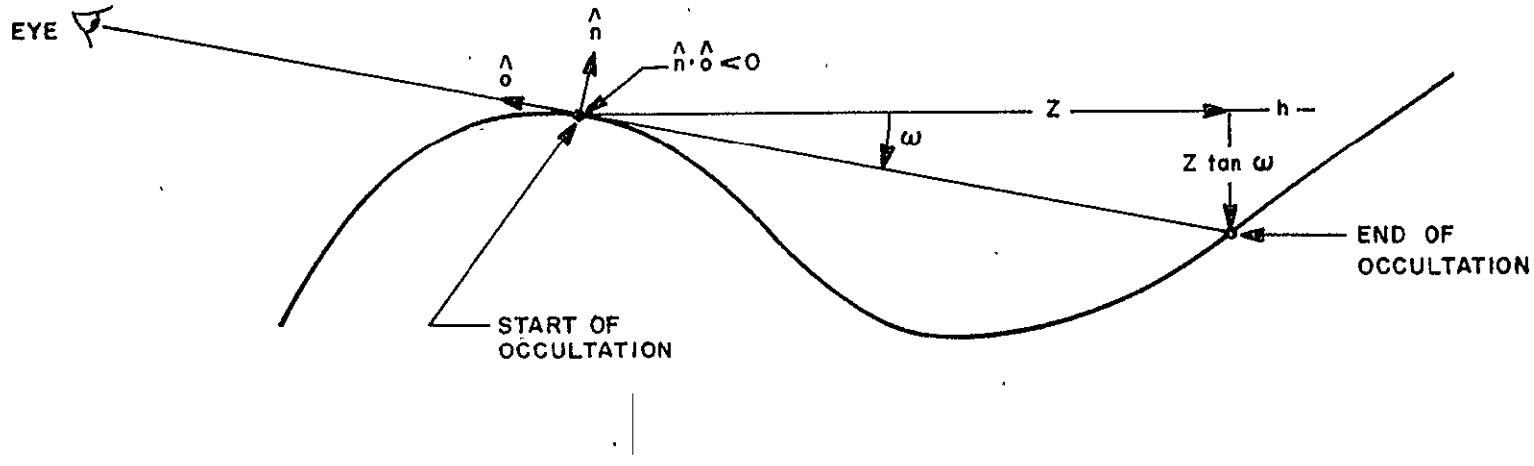



FIG. 8 - GRAPHIC ILLUSTRATION OF OCCULTATION

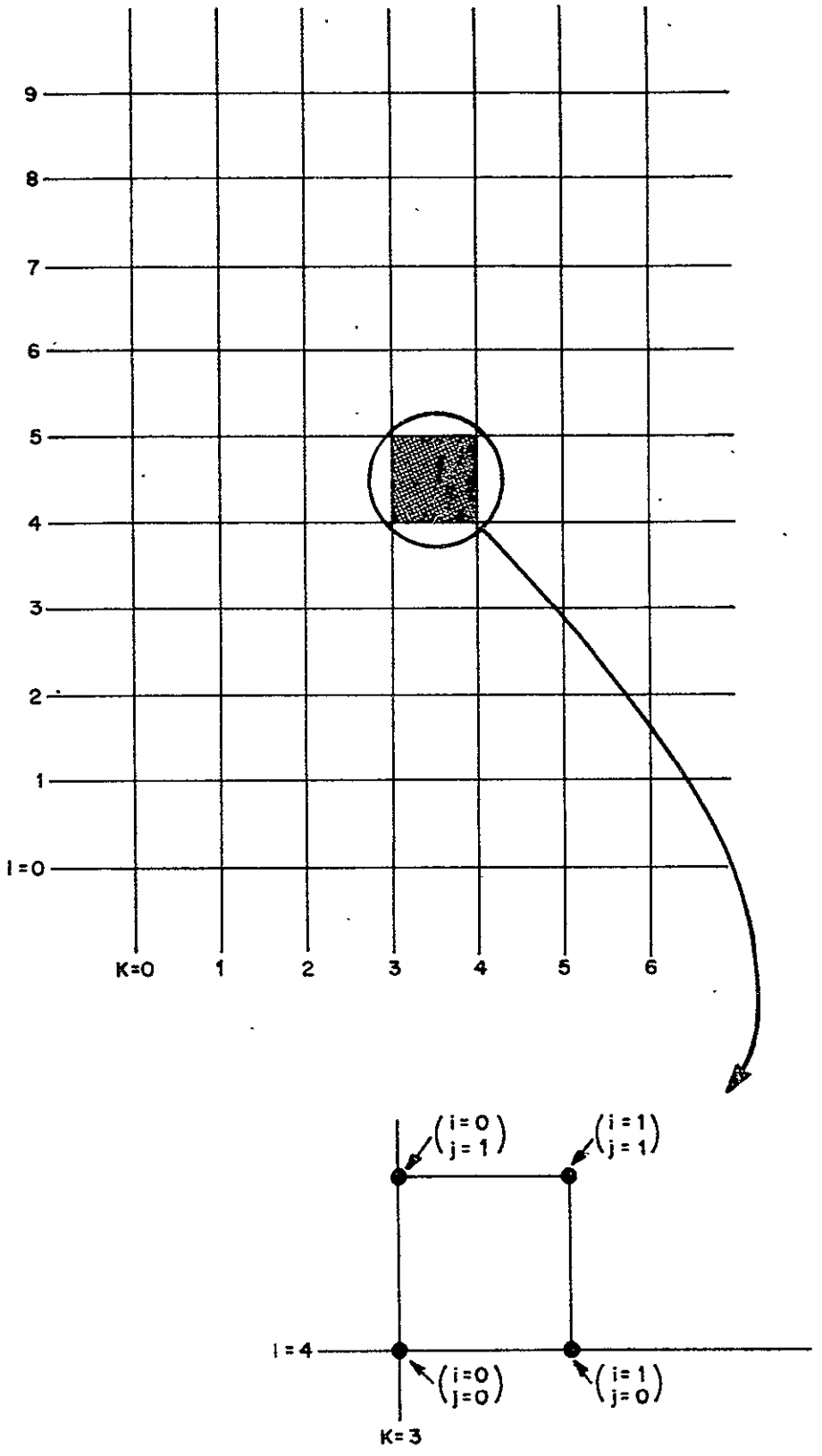



FIG. 9 - REGION COORDINATES

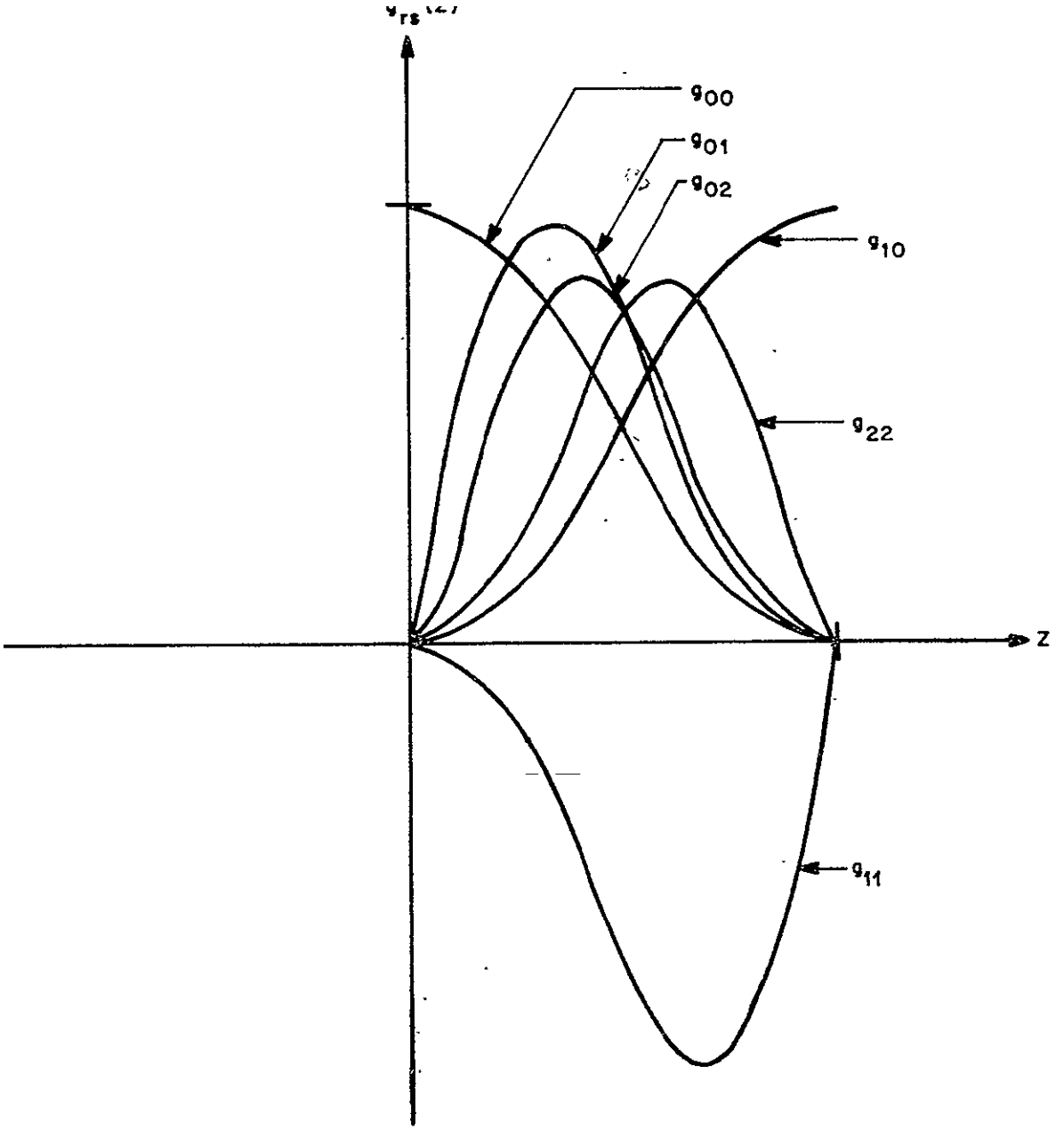



FIG. 10 - POLYNOMIAL TERMS

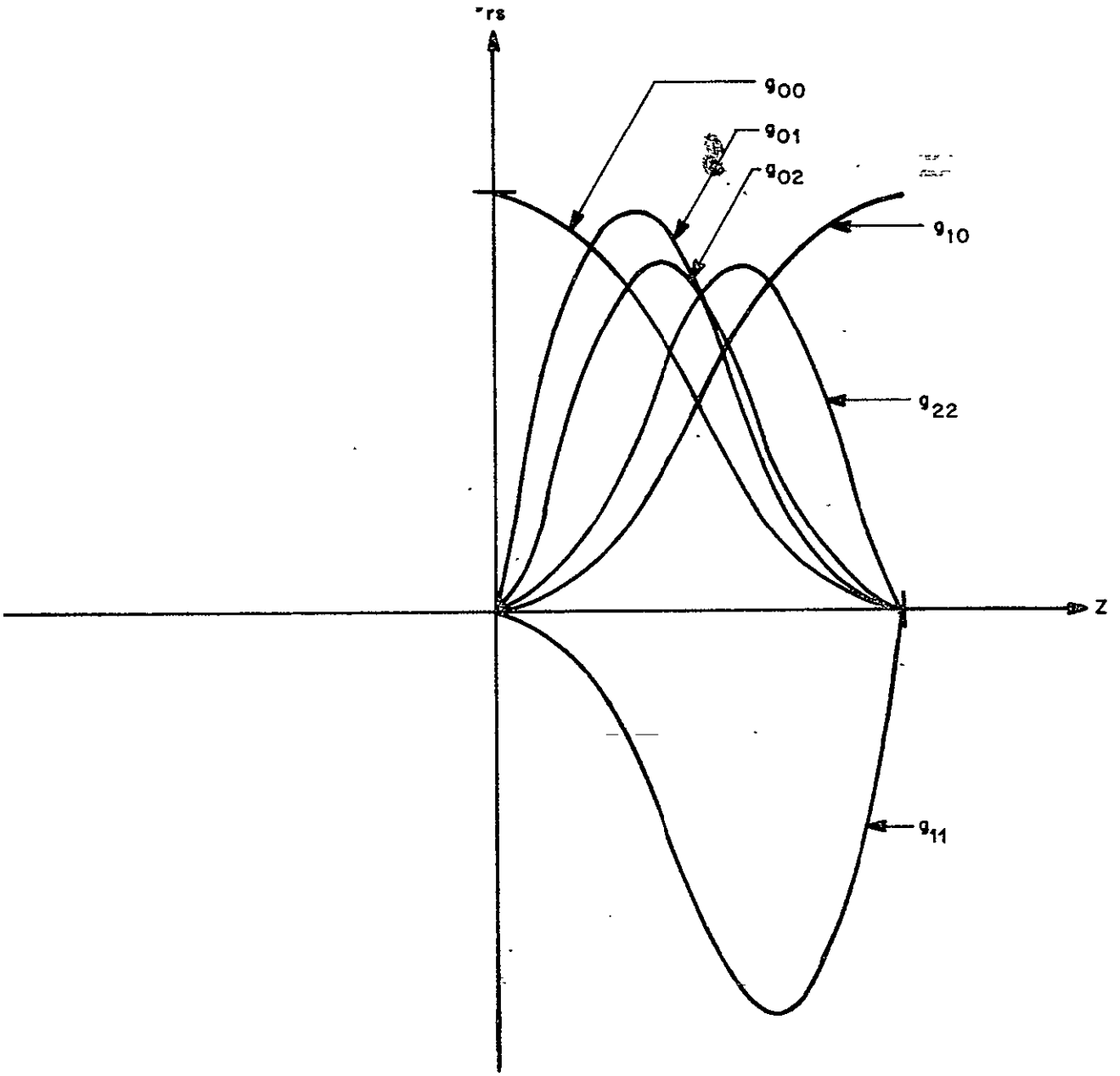



FIG. 10 - POLYNOMIAL TERMS

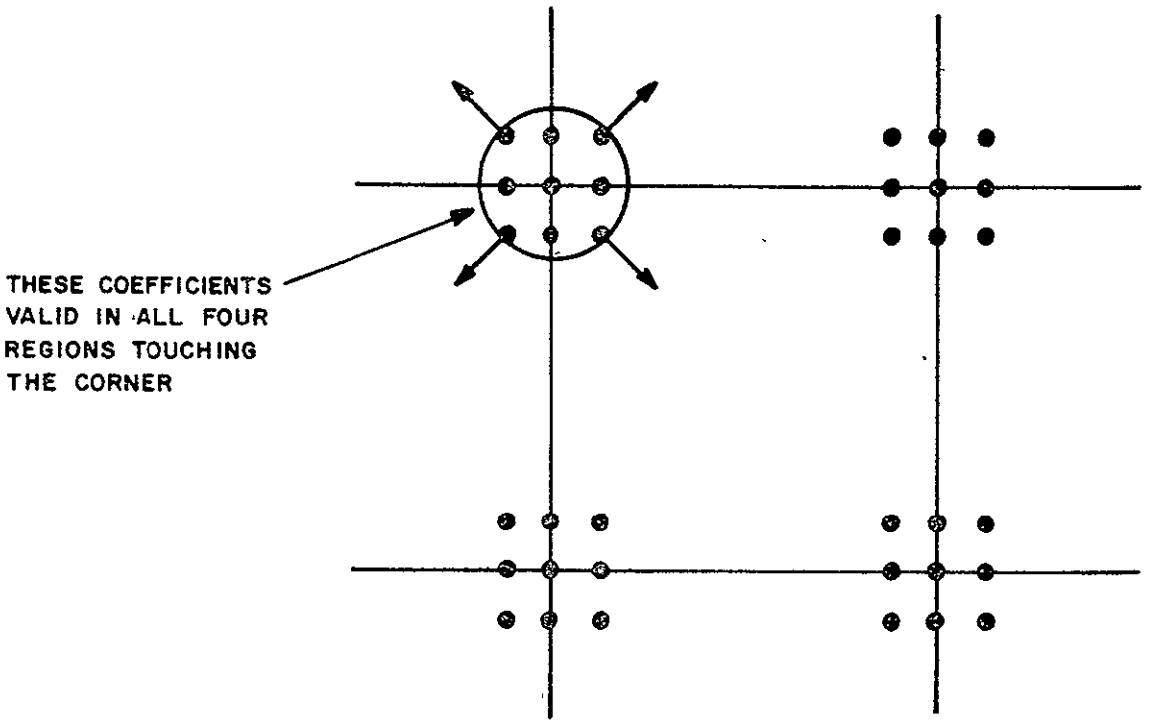
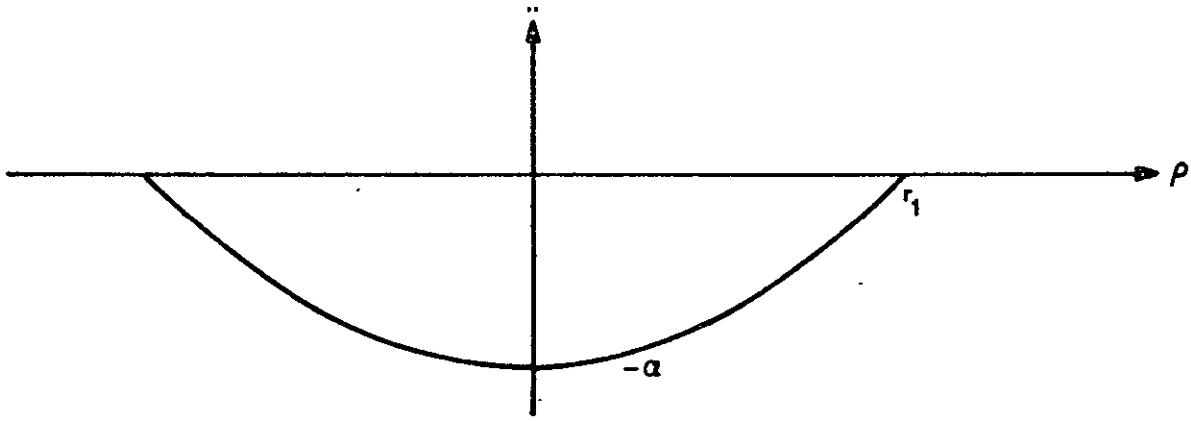
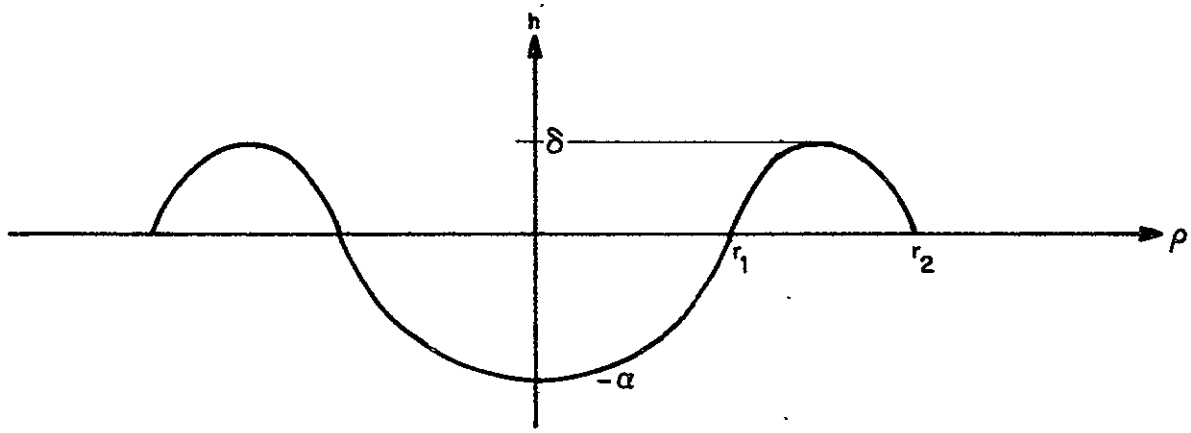


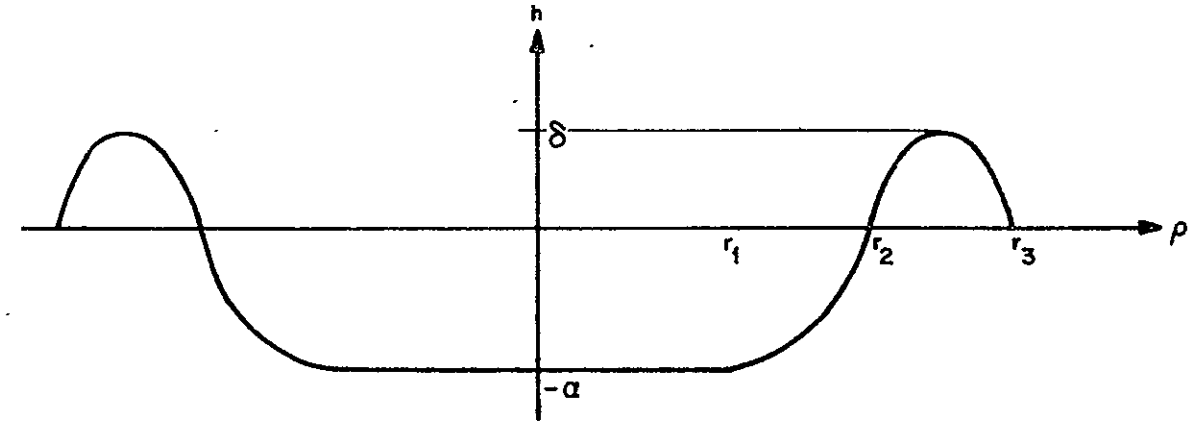

FIG. II - COEFFICIENTS USED FOR ONE REGION




(a) TYPE 1 CRATER FORM




(b) TYPE 2 CRATER FORM




(c) TYPE 3 CRATER FORM

FIG. 12 – SPECIAL FUNCTION REPRESENTATION OF CRATERS

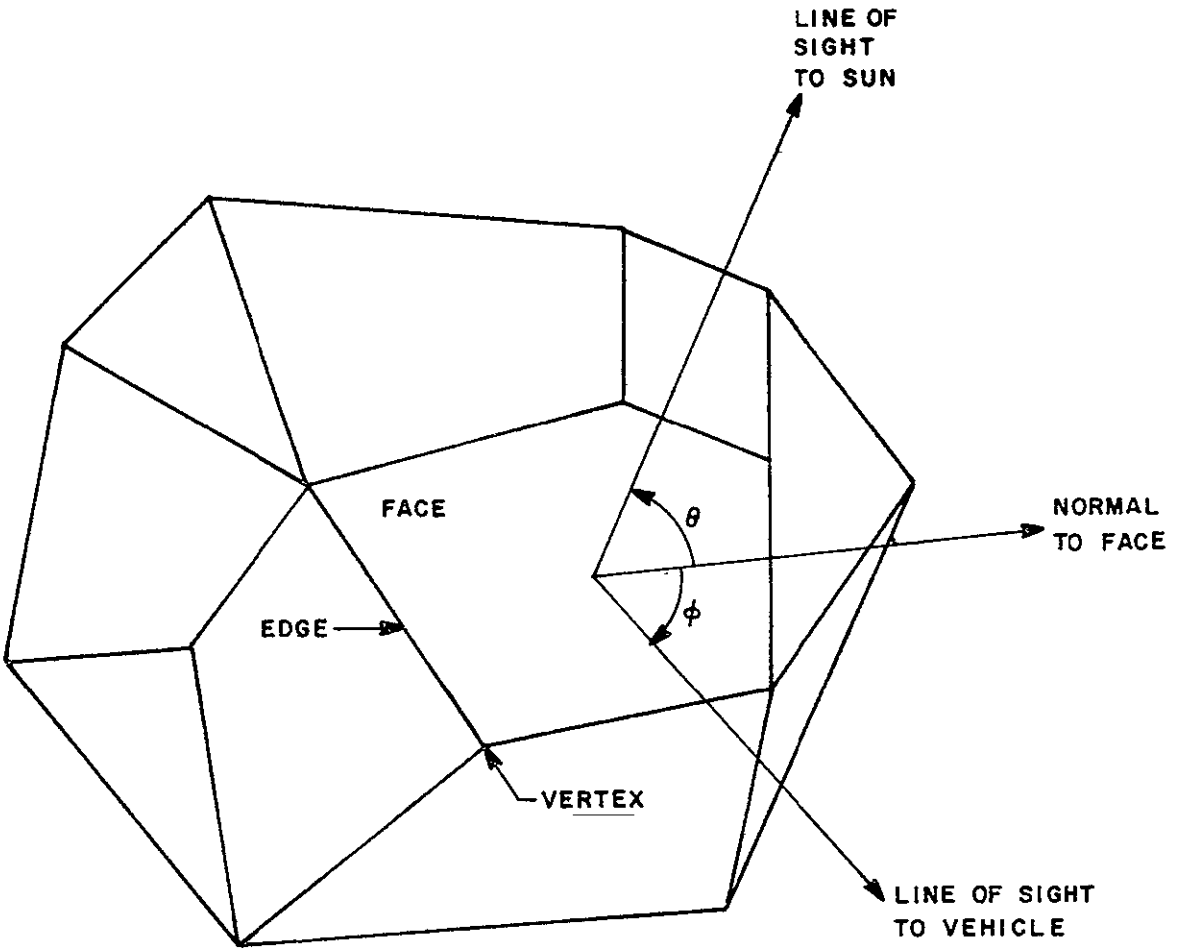



FIG. 13 – ILLUSTRATION OF A ROCK AS A POLYHEDRON

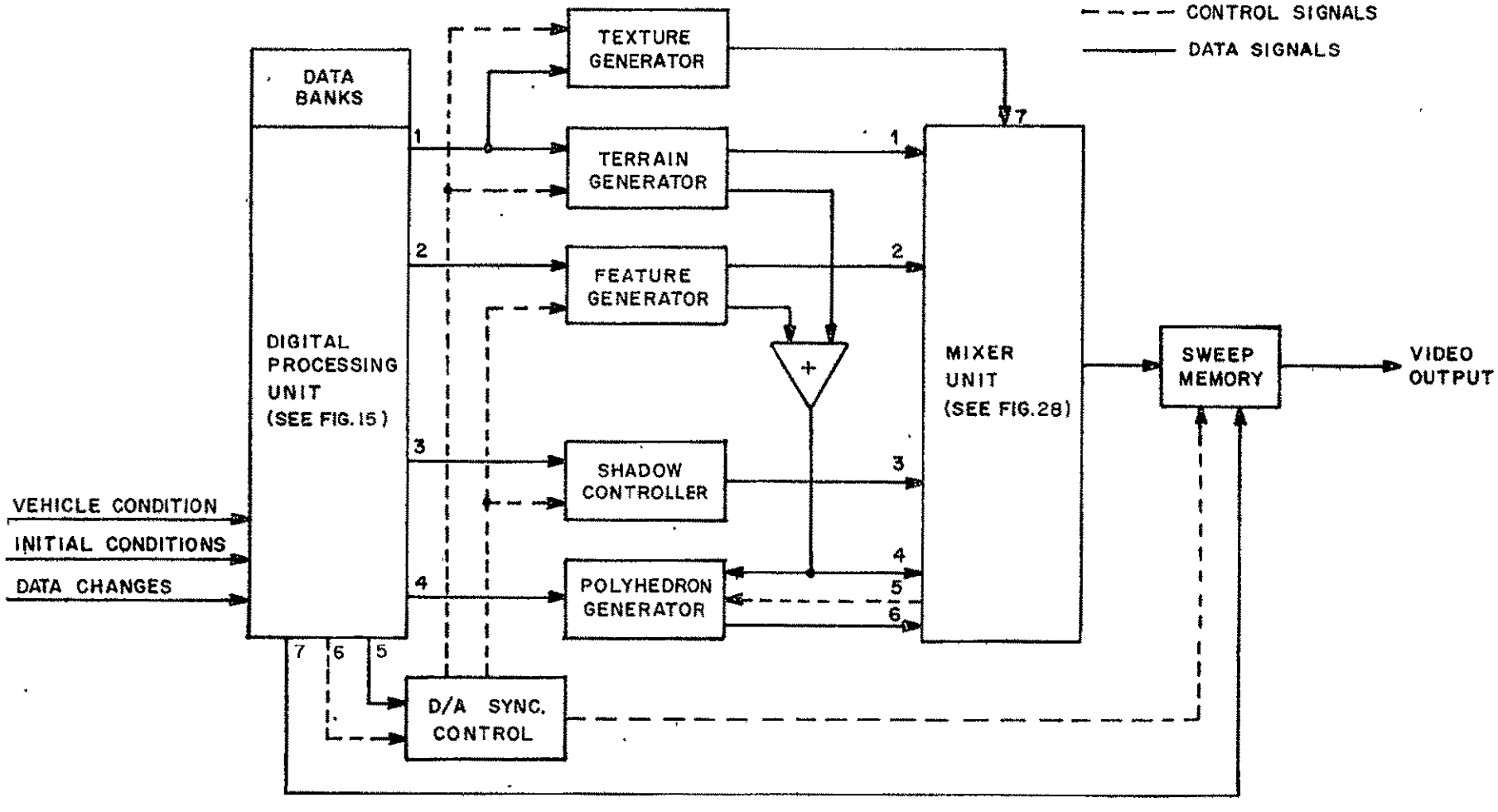



FIG. 14 – OVERALL SYSTEM CONFIGURATION

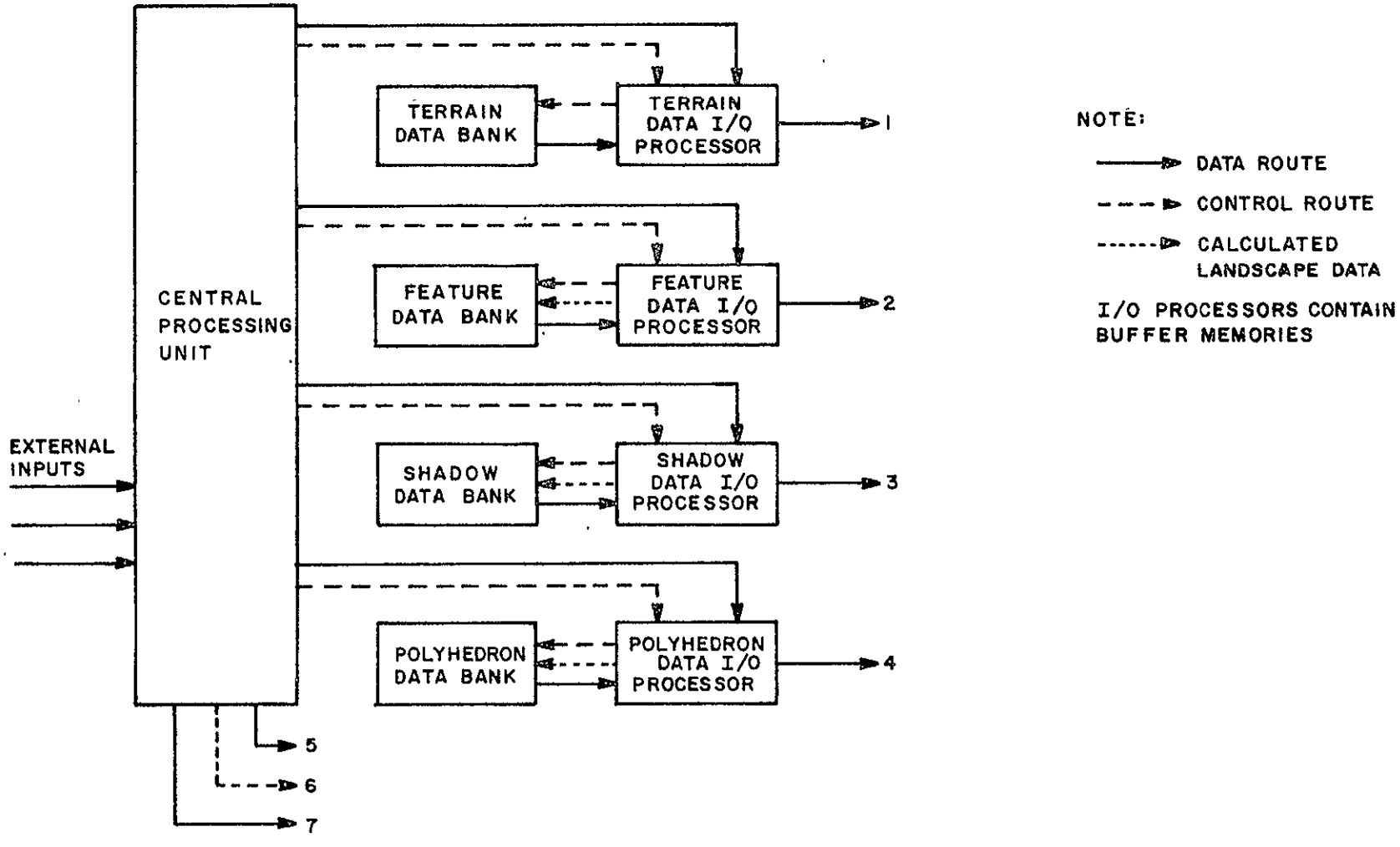



FIG. 15 – ORGANIZATION OF DIGITAL PROCESSING UNIT (DPU)

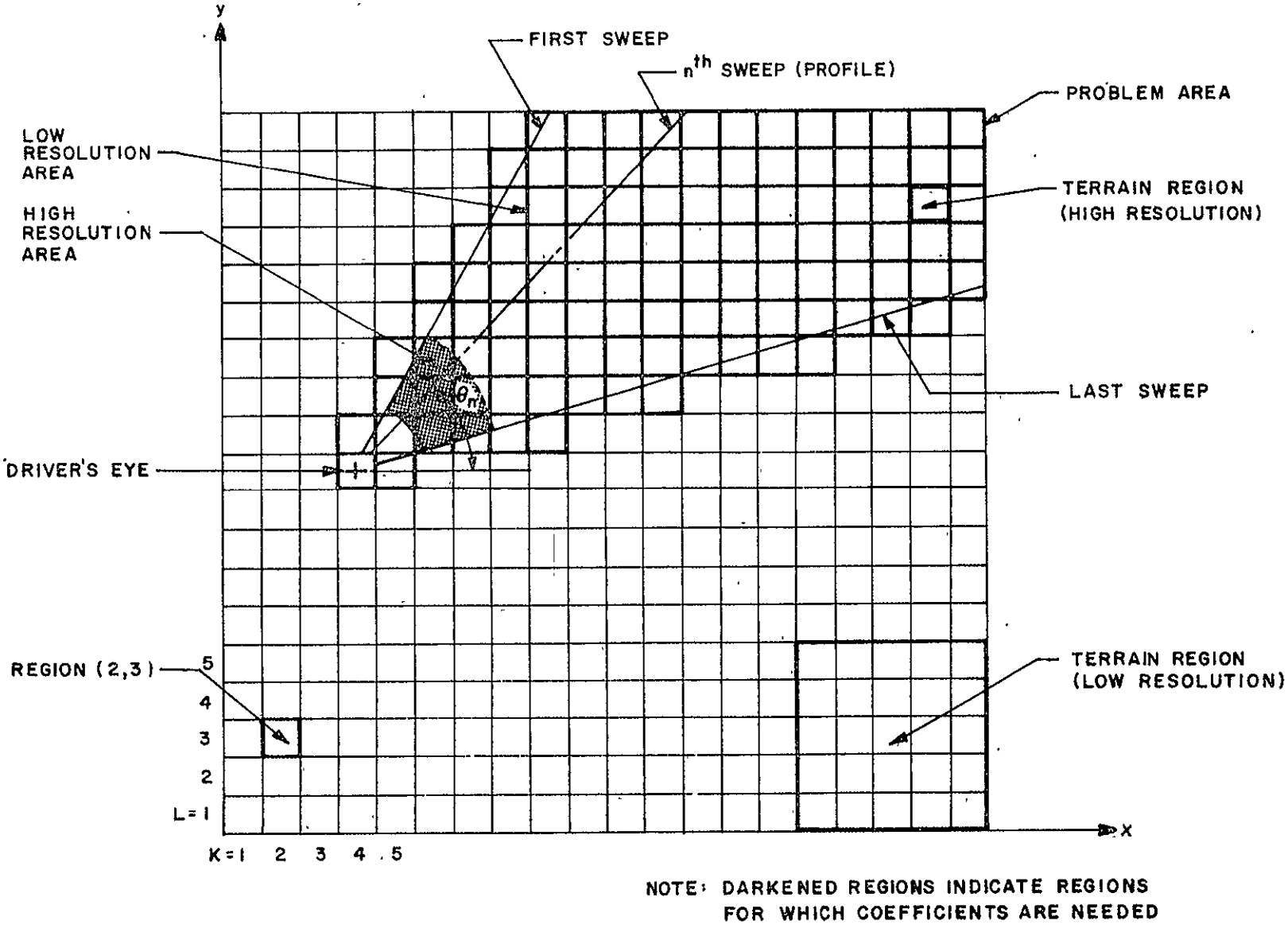



FIG. 16 – POLYNOMIAL REGIONS

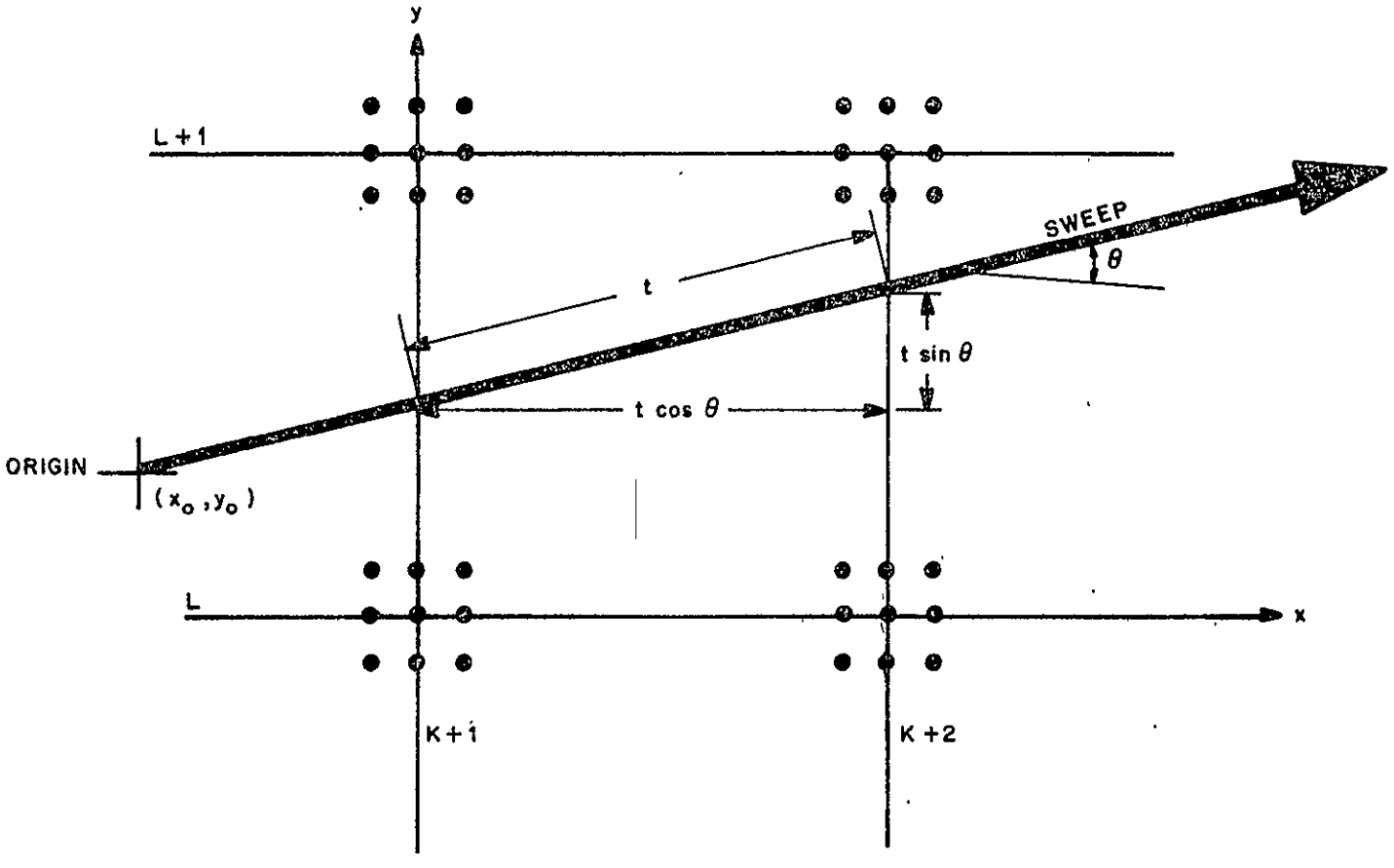



FIG. 17 – SWEEP THROUGH ONE REGION

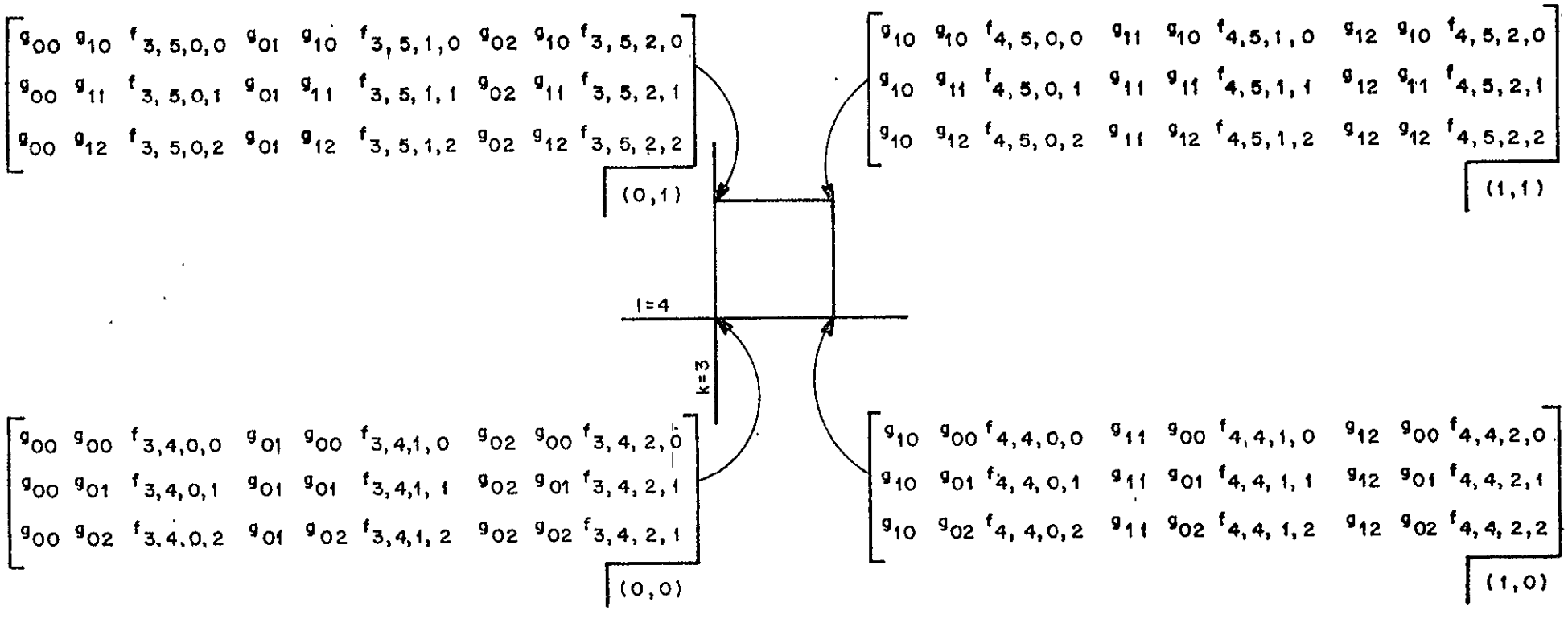



FIG. 18 – COEFFICIENT PLACEMENT

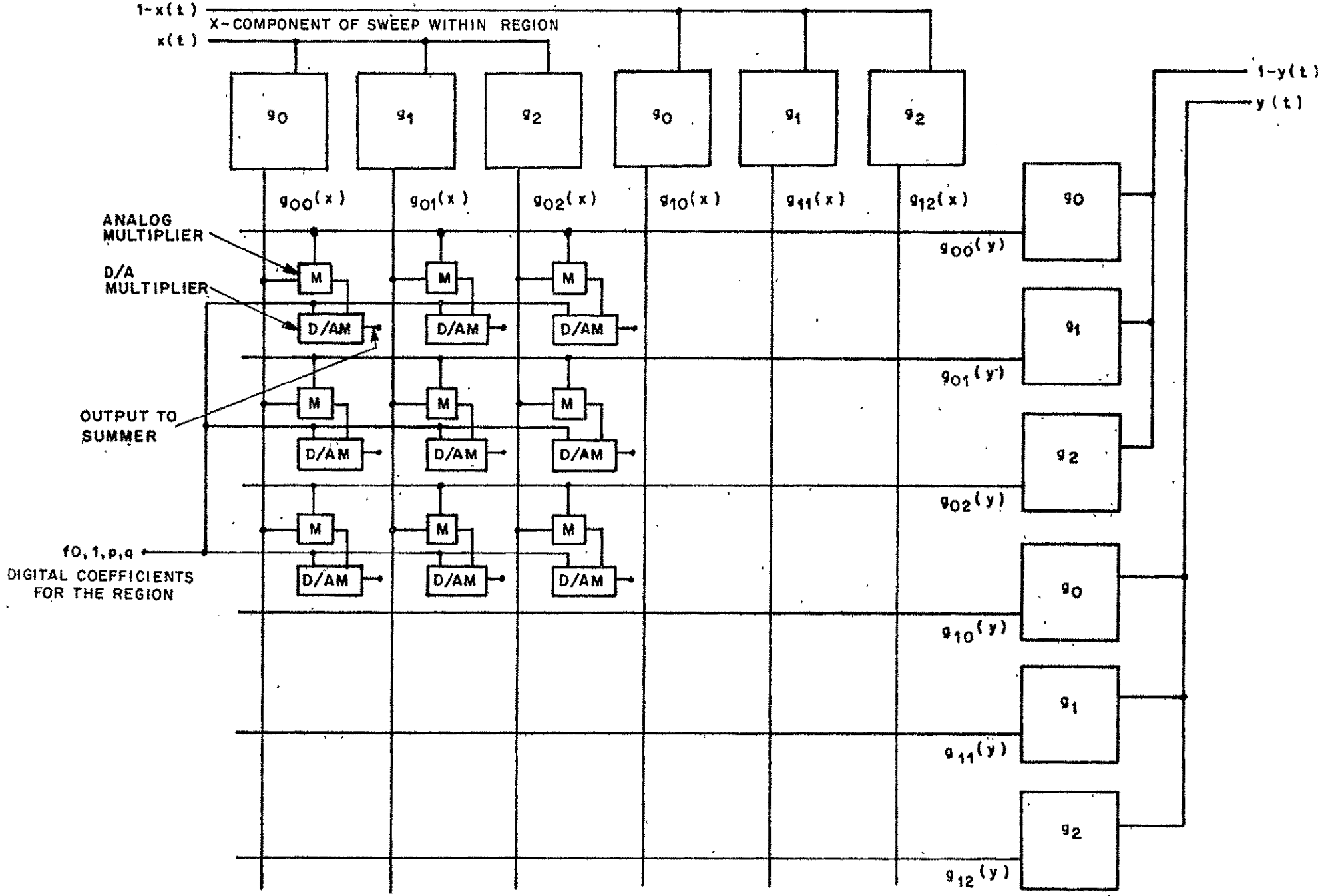



FIG. 19 - EQUIPMENT CONNECTIONS

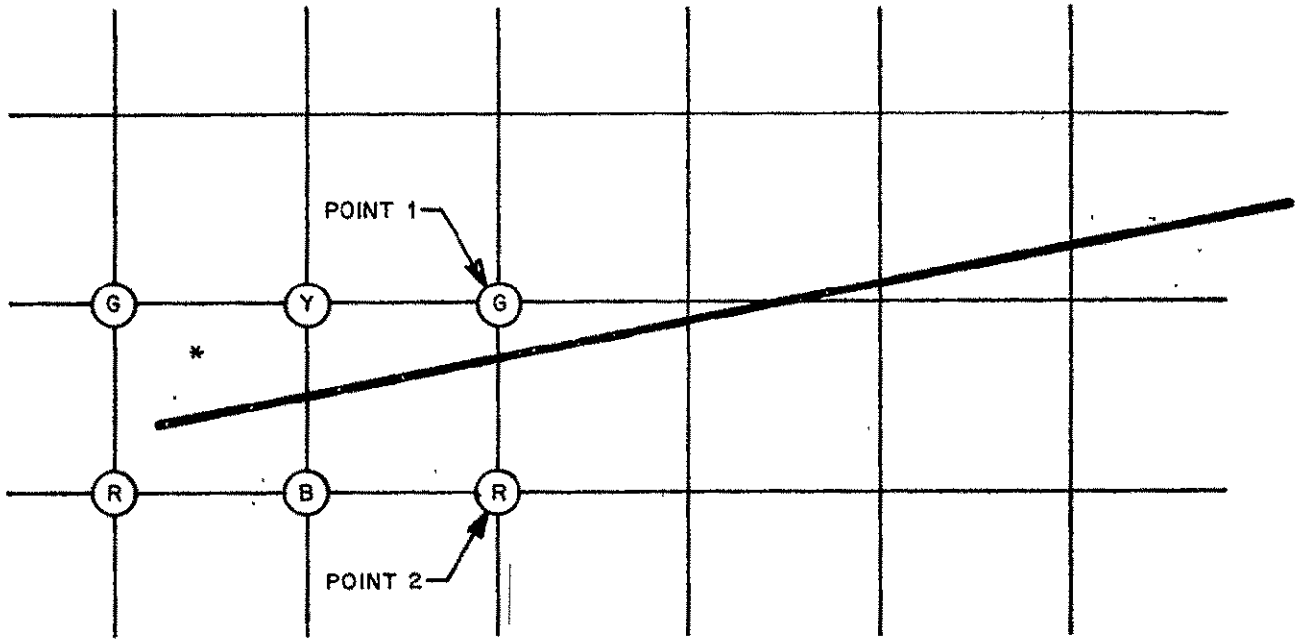



FIG. 20 – ORDERING OF THE COEFFICIENTS

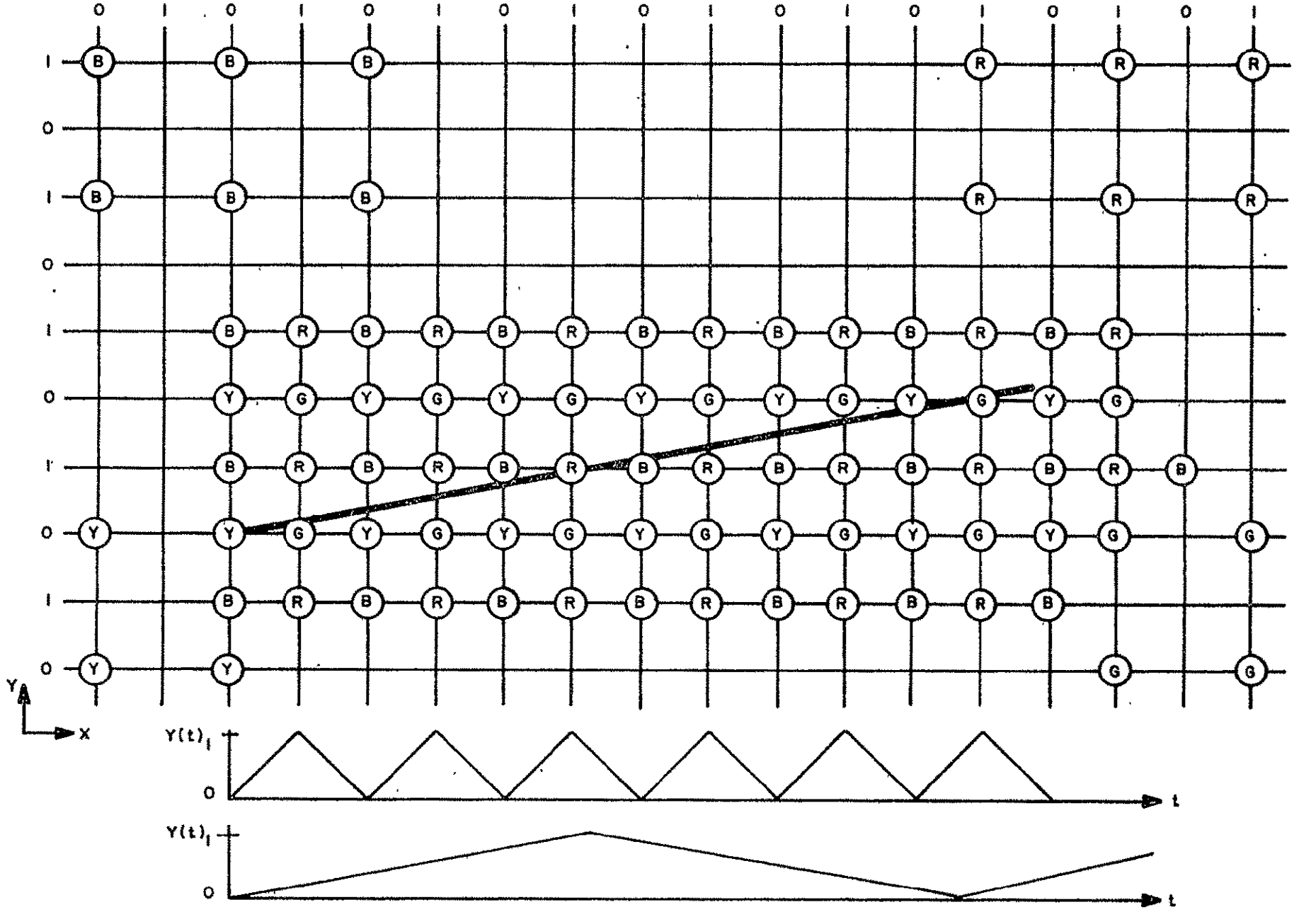



FIG. 21 - COEFFICIENTS ON FOUR INTERLACED ARRAYS

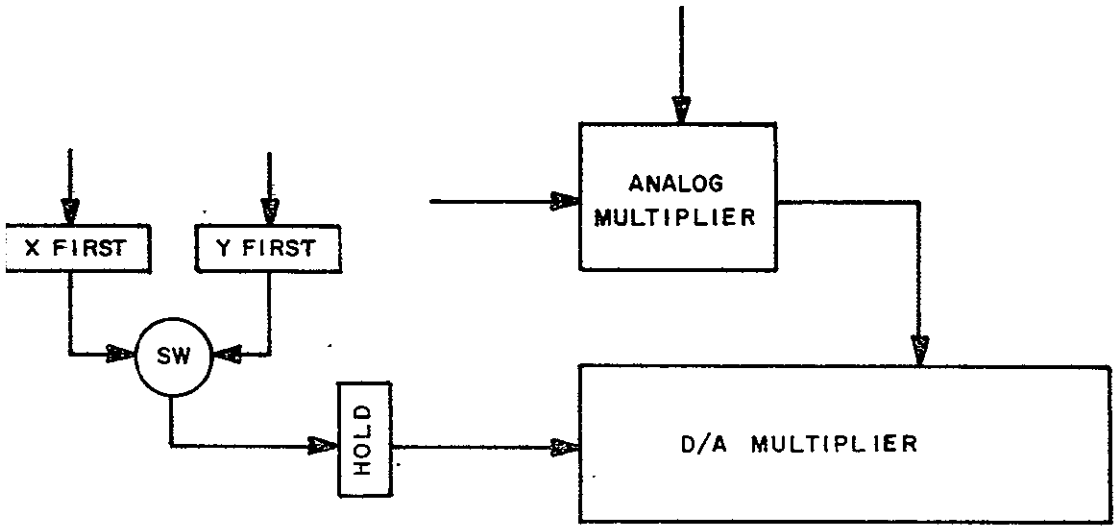



FIG. 22 – SWITCHING OF COEFFICIENTS

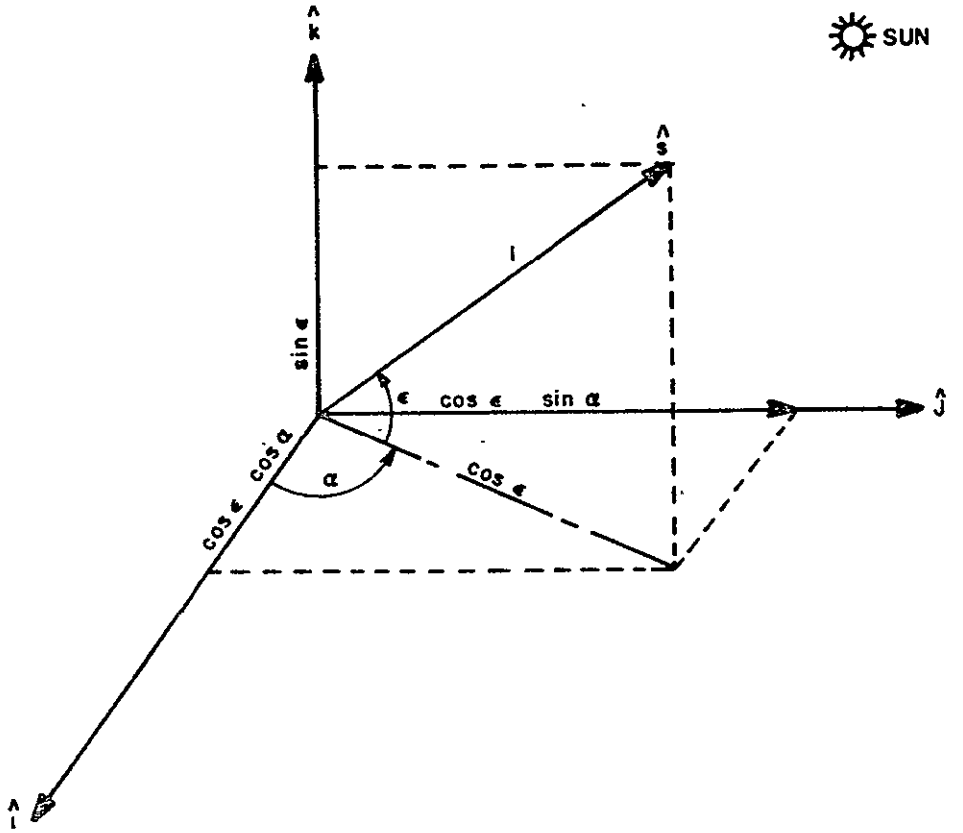



FIG. 23 – COMPONENTS OF UNIT VECTOR TO SUN

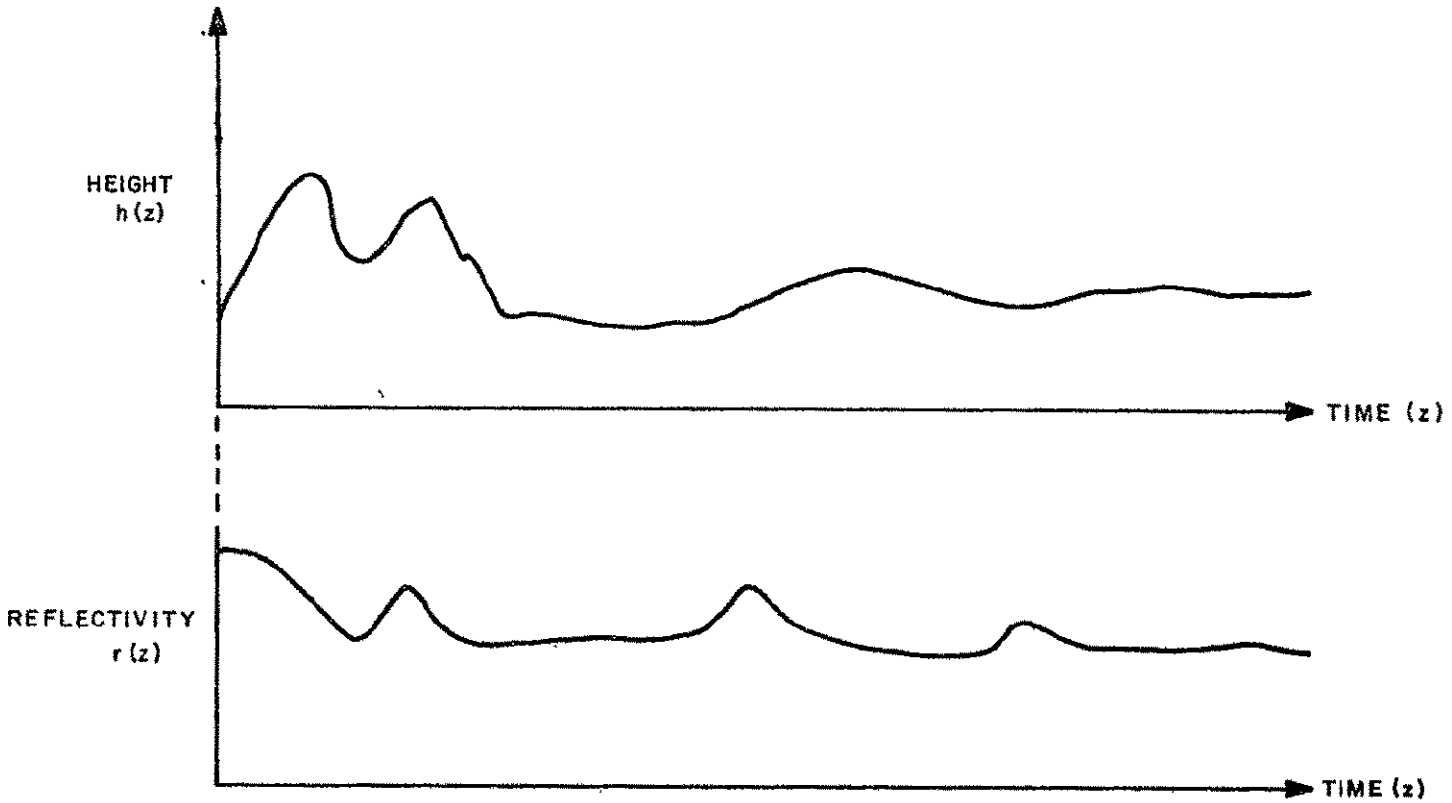



FIG. 24 – TYPICAL HEIGHT AND REFLECTIVITY PROFILE SIGNALS

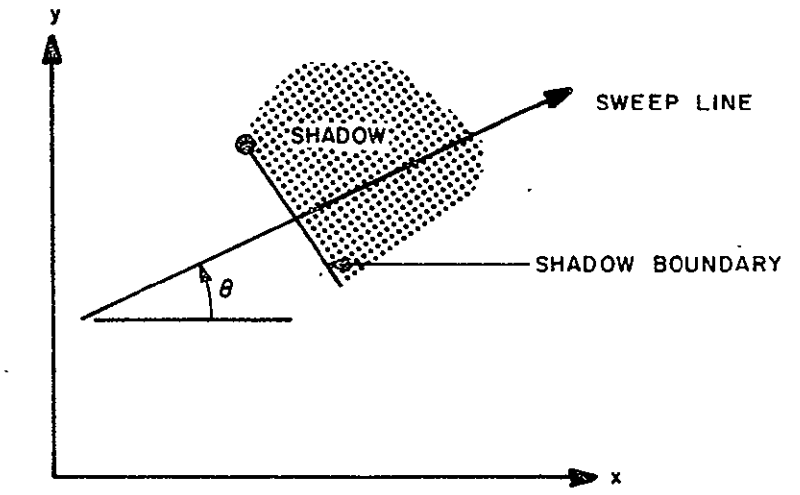



FIG. 25 - CALCULATION OF SHADOW BOUNDARY CROSSINGS

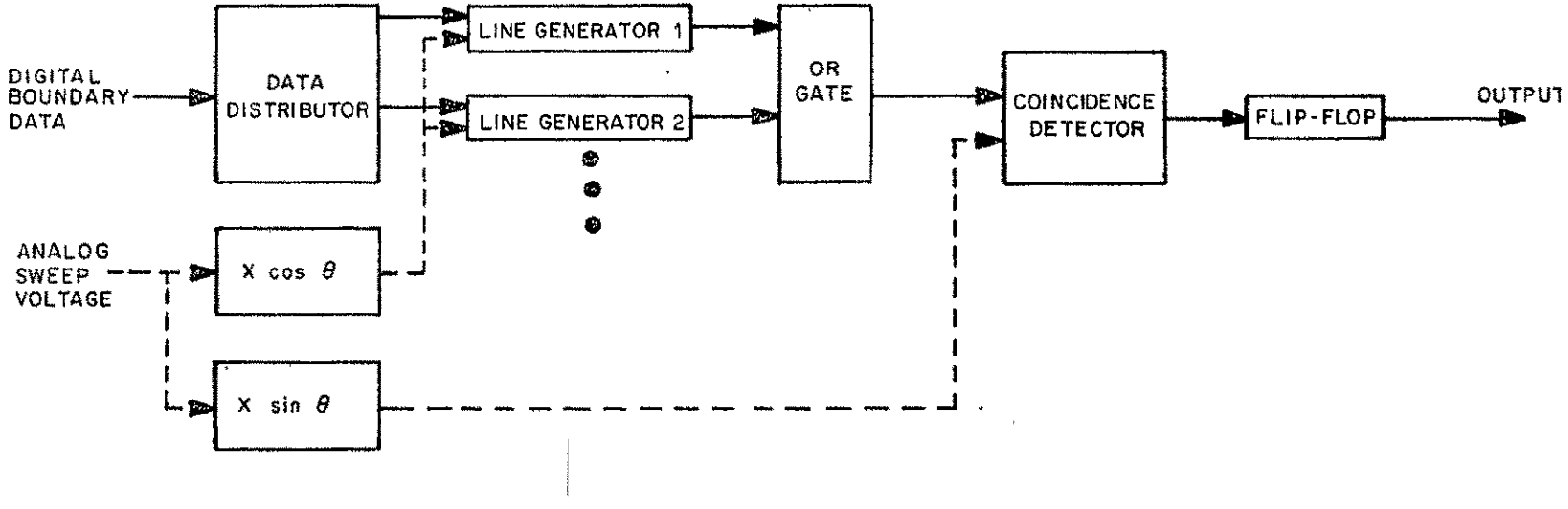



FIG. 26 - DIAGRAM OF DIGITAL ORIENTED SHADOW CONTROLLER

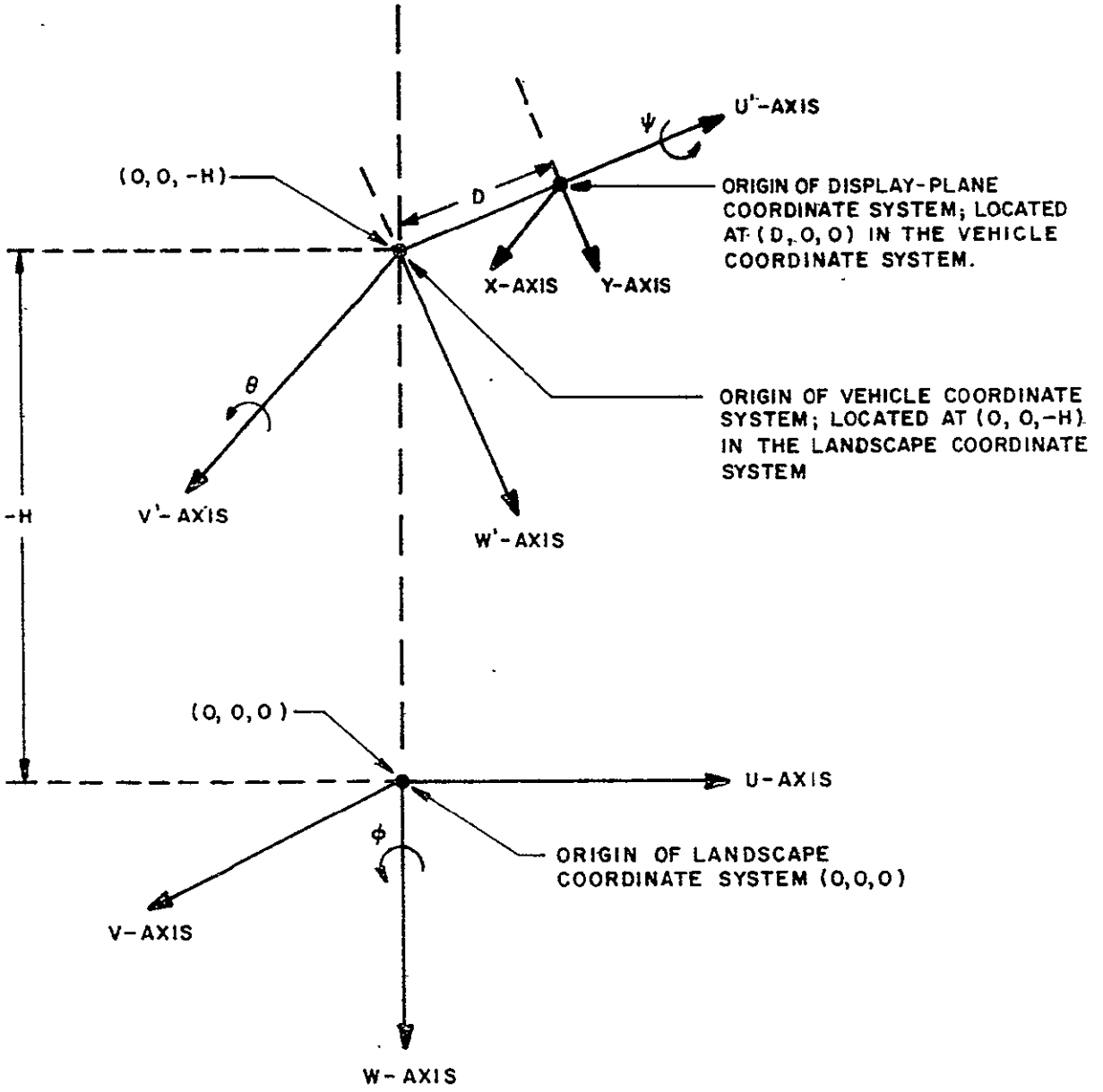



FIG. 27 - RELATIONSHIP OF THE COORDINATE SYSTEMS

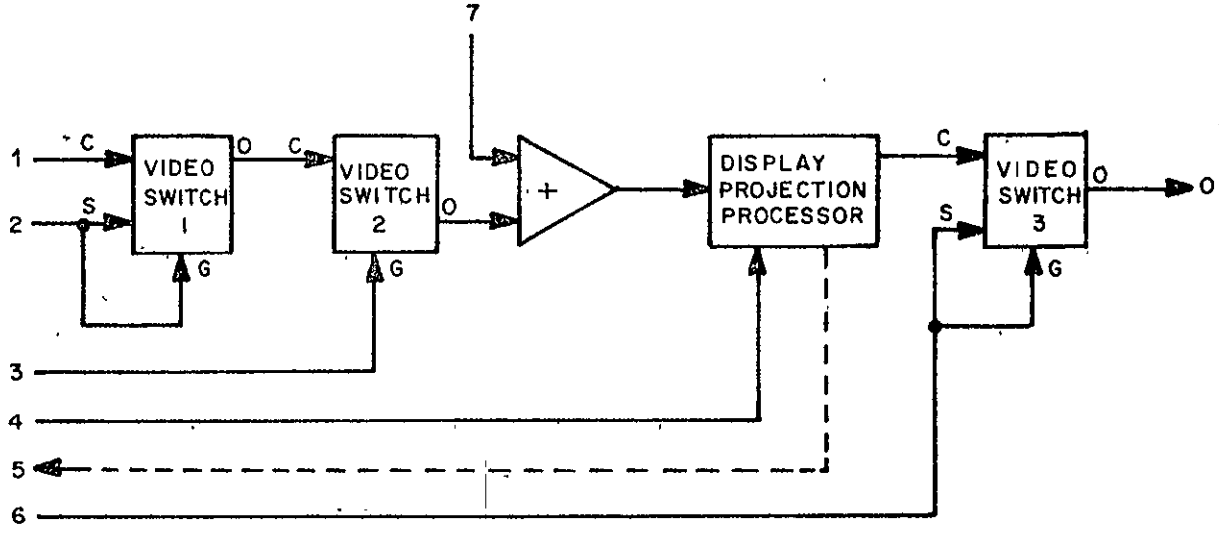



FIG 28 - DETAIL OF MIXING UNIT

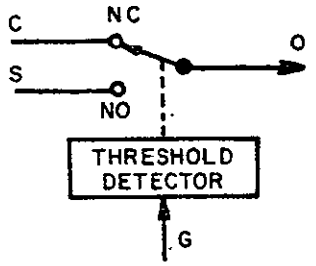



FIG. 29 – DETAIL OF VIDEO SWITCH

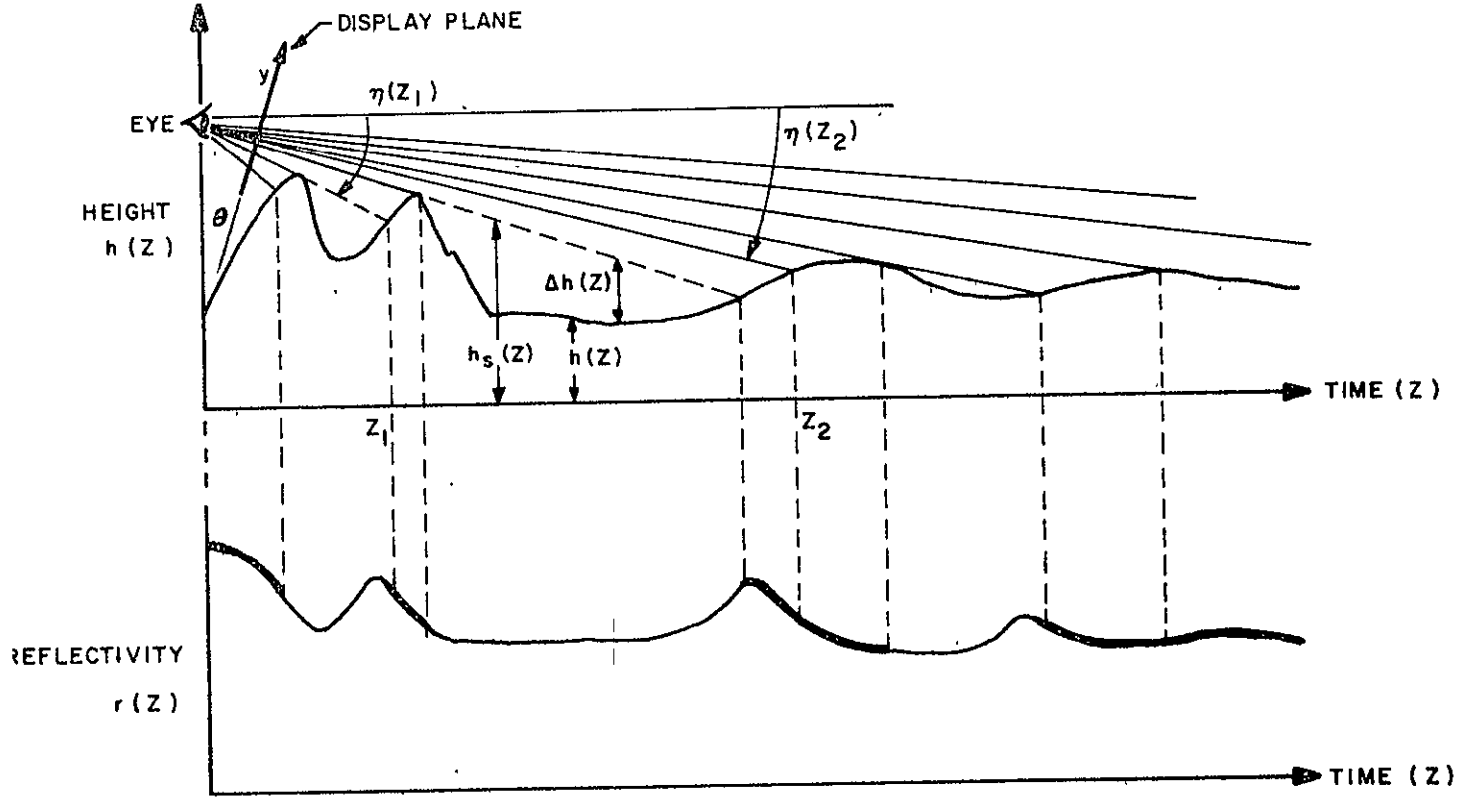



FIG. 30- DISPLAY PROJECTION OF REFLECTIVITY PROFILE

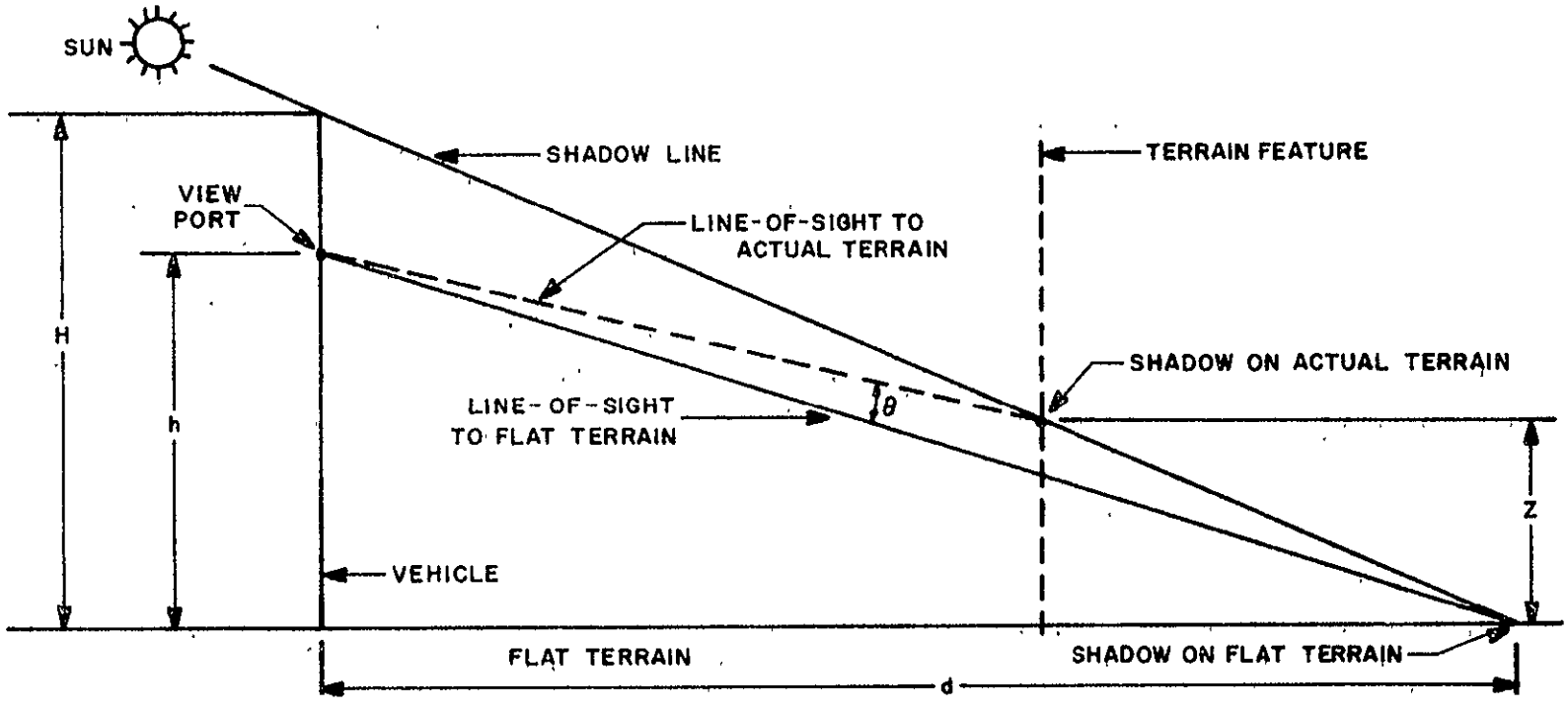



FIG. 31 - VEHICLE SHADOW ON FLAT TERRAIN

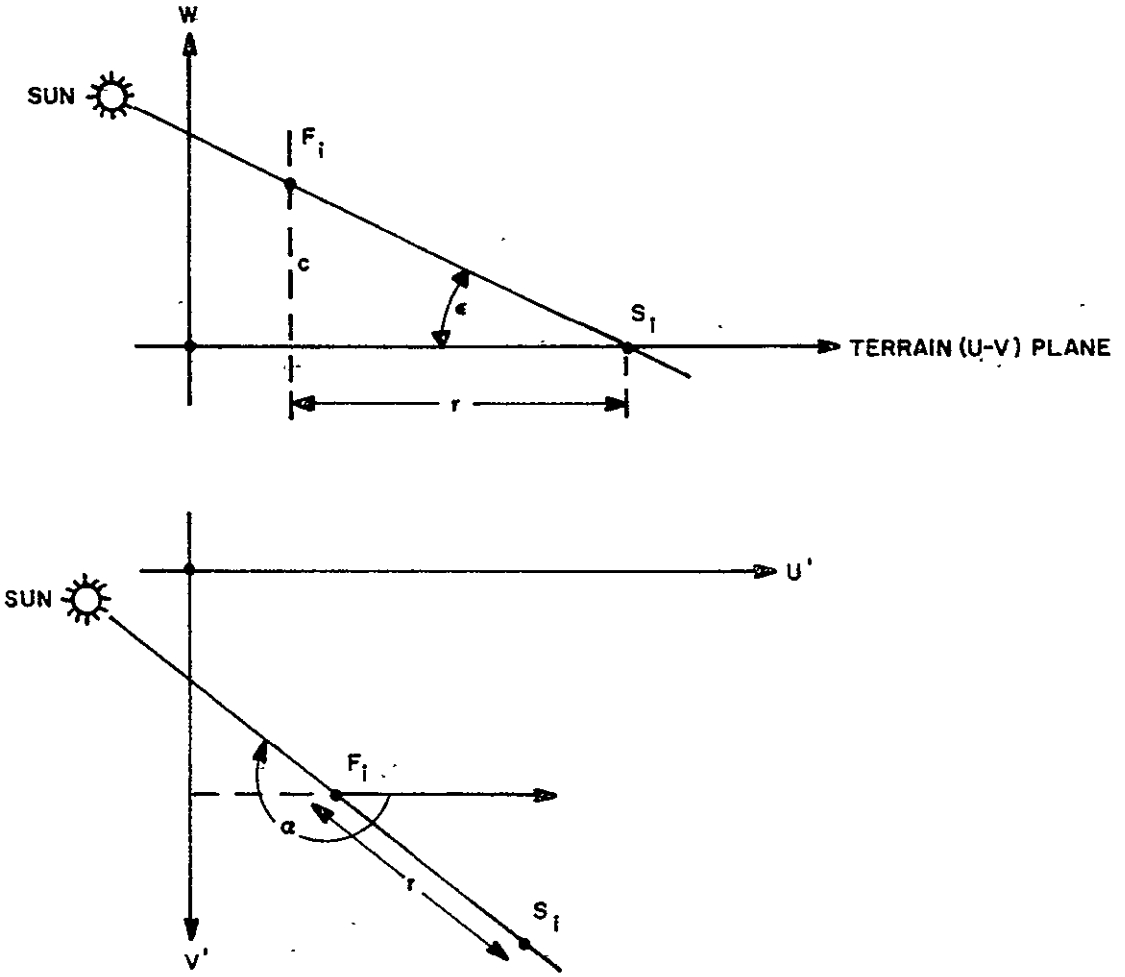
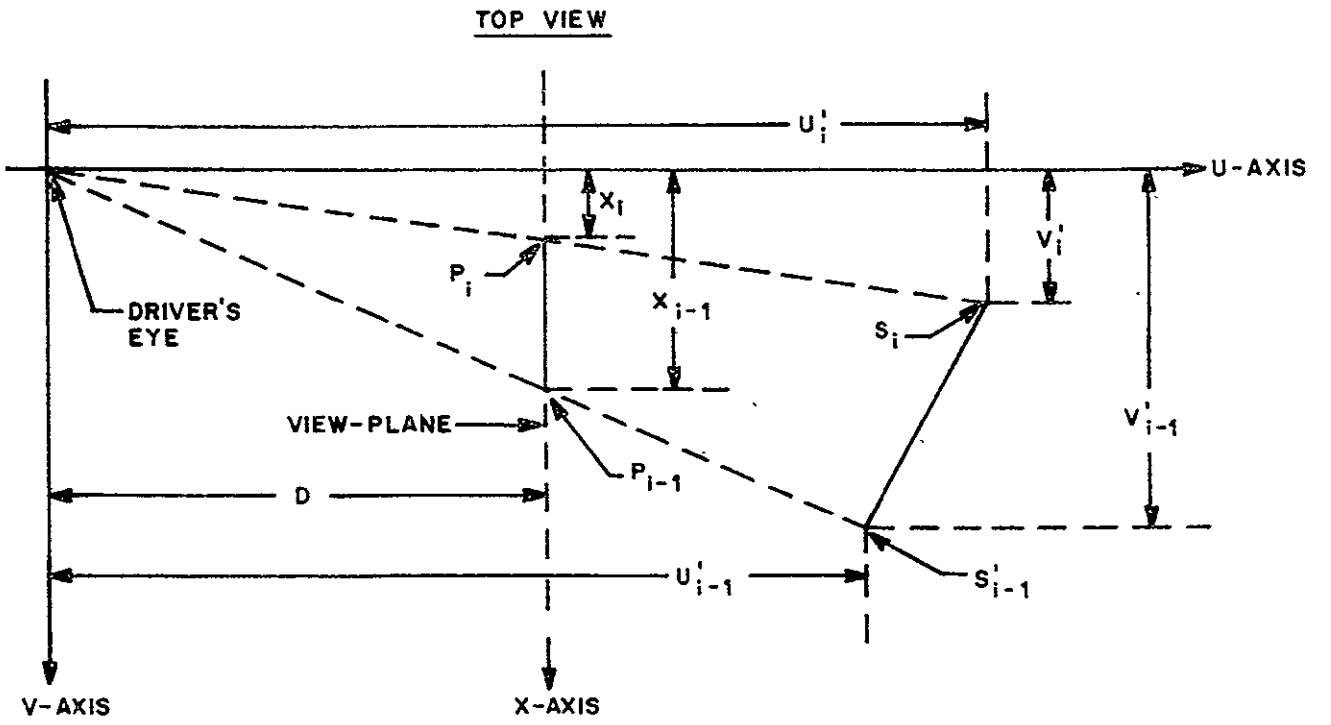
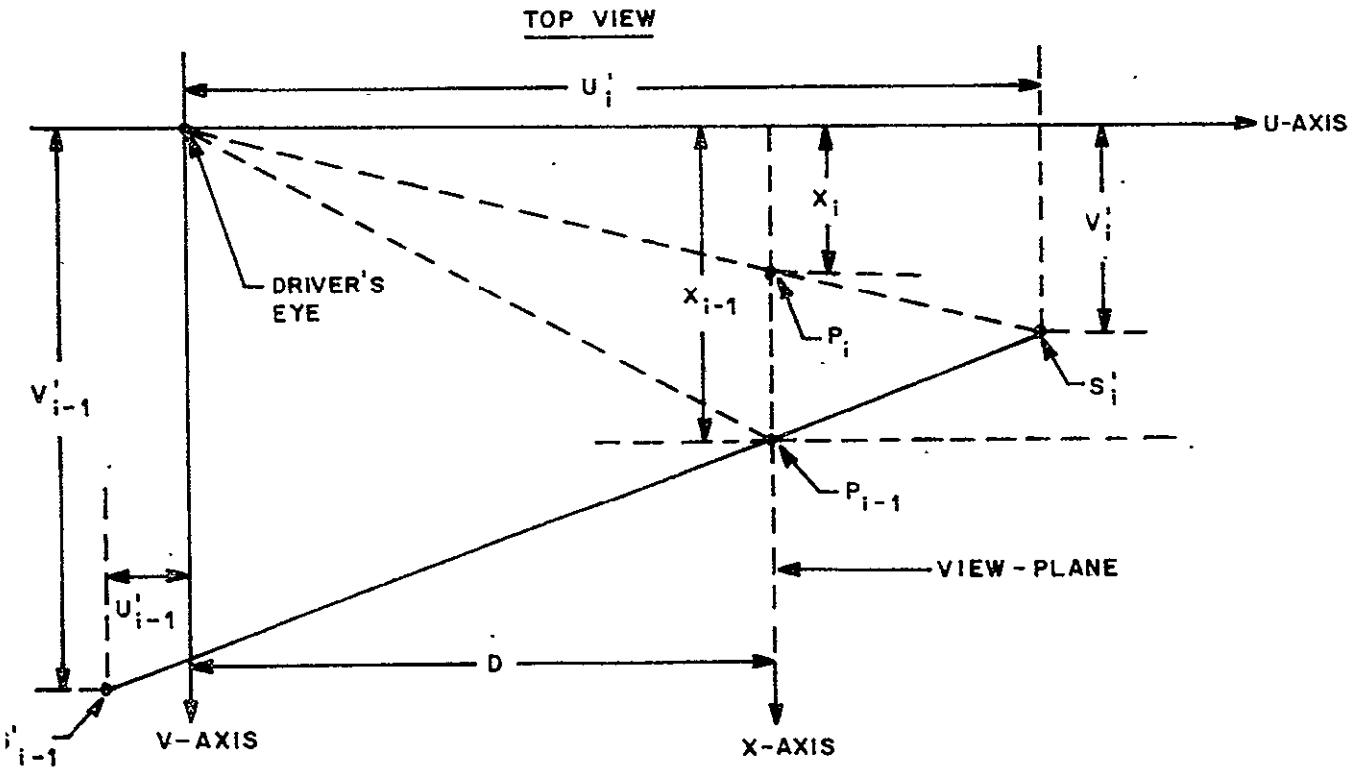



FIG. 32 – SHADOW CALCULATION




(a) VIEW-PLANE PROJECTION OF TYPE I POINTS




(b) VIEW-PLANE PROJECTION OF TYPE 2 POINTS

FIG. 33 – VIEW-PLANE PROJECTIONS

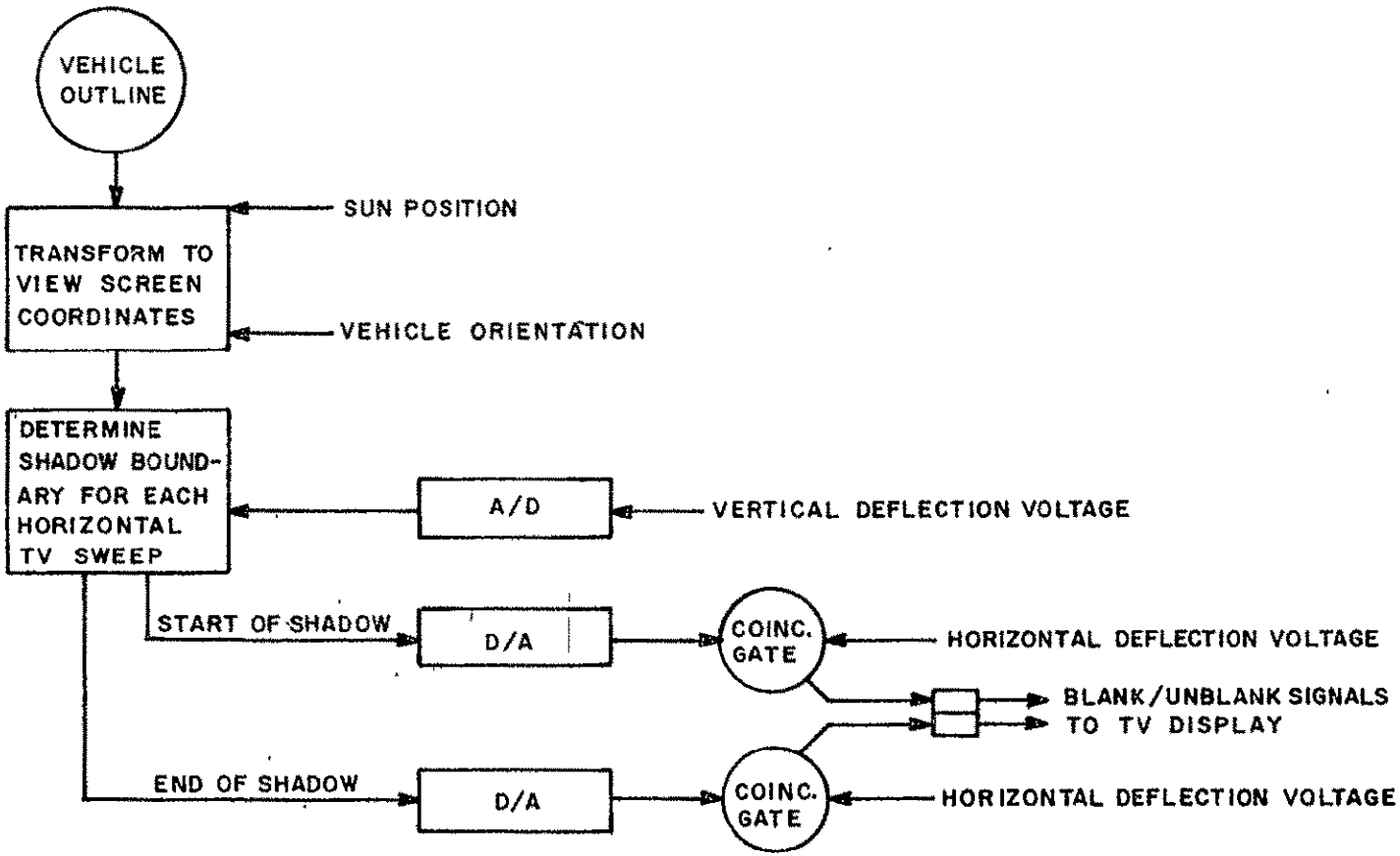
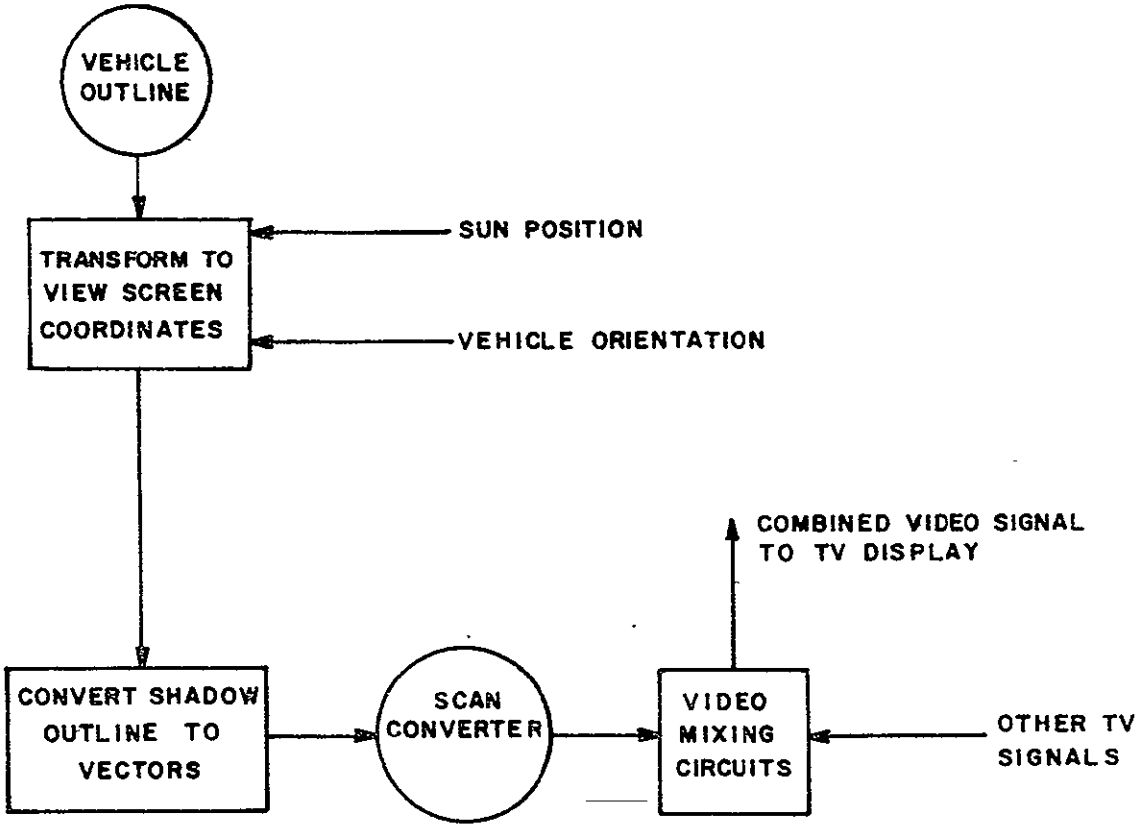



FIG. 34 – SYNCHRONIZED METHOD




NOTE: THE SCAN CONVERTER CONVERTS VECTOR GENERATED SHADOWS TO RASTER SCAN SIGNAL

FIG. 35 - USE OF SCAN CONVERTER

# APPENDIX:
# SIMULATION OF VISUAL DISPLAYS BY GENERAL-PURPOSE COMPUTING EQUIPMENT

In order to demonstrate the feasibility of creating realistic visual simulation of the moon's surface with a digital computer, Pennsylvania Research Associates Inc. has developed a Lunar Simulation Program (LUNSIM). This program was run on an IBM 360/65, producing magnetic tapes with instructions for a Stromberg-Carlson 4020 Computer-Recorder. Several series of photographs were made from the 35 mm positive film output of the SC 4020. Three of these sequences appear as Exhibits A, B, and C of this report. Although these pictures were not produced in real time they nevertheless use many of the same techniques developed for a real-time simulator. Of particular importance, is the use of conceptual crater generators, or pieces of hardware that implement standardized formulas to represent crater shape.

One of the prime features of the LUNSIM system is that the data for the lunar surface is kept in a compressed form. Instead of storing heights for every point on the surface LUNSIM takes as input a small number of parameters describing craters, which are completely relocatable. In a sense, a "crater generator" is included in the program, using analytical functions to create surface variations. A description of LUNSIM follows.

LUNSIM is a FORTRAN and Assembly Language program which generates display instructions for producing pictures of the moon's surface on the SC 4020 Computer-Recorder. The pictures produced are simulated views of small portions of the lunar landscape as seen by a landing vehicle. A rather simplified model of the moon was created in order to reduce the program running time and memory requirements to economically feasible amounts. The major features and restrictions of the model are:

- The surface is perfectly smooth except where individual features (craters) occur.

- Each individual feature is one of a small number of types of features. In the current version of LUNSIM, features are three types of craters, all of which are symmetrical (i.e., the crater height values depend only on the distance from the crater center).

- The observer (in a landing vehicle) must have an azimuthal position on one of the four cardinal points of the compass. He may, however, be located over any ground position.

- Altitude is virtually unlimited and elevation angle can be between zero and ninety degrees. The limitation on the vertical field of view angle is such that the upper line of sight does not extend above the horizontal.

- The sun must be on one of the four cardinal points of the compass. The solar elevation angle can be between zero and ninety degrees.

The inputs to the program are of two types:

- Parameters placing the observer and sun positions and angles (known as run parameters).

- Crater data, consisting of 2 to 5 parameters for each crater, describing the type, size, and shape, and the ground coordinates of the crater center.

The crater data was obtained by making the appropriate measurements on Lunar Map ORB-II-6 (100) which was prepared from Lunar Orbiter II photographs. Approximately 400 craters were coded and keypunched for a 43 km × 43 km square contained with the map. All calculations -- heights, slopes, intensity returns -- are done on a point-by-point basis on a cartesian grid. The grid size is a run parameter, limited only by the display screen, in this case the 1024 × 1024 plottable positions on the SC 4020.

The following description of LUNSIM refers to the macro-flowchart given in Fig. 1. The operations denoted in blocks 1-8 of the flow-chart are the precalculation stage of the LUNSIM instruction-generating process. Since the program was written to accept many combinations of observer and sun positions, the logical paths followed by the generalized calculation subroutines (blocks 10-15) are set (at block 2) after the run parameters are read.

Various constants (which are functions of the run parameters) are pre-calculated next. These constants and all precalculated information are communicated to the several subroutines via storage labelled COMMON. Having all the information describing the observer's position and attitude, the program now determines what portion of the lunar surface (map) will be viewed (block 5). Next the crater data is read in and only parameters for craters which fall within the viewed area are saved. Two tables are generated, containing these parameters and the coordinates of the bounding square for each usable crater (block 7).

All calculations are done on a row or column of ground coordinates at a time. A special algorithm treats the "leading" edge of the area which lacks an adjacent row of heights on one side (block 9). The main loop consists of blocks 10-17. Heights are generated for a row by finding which craters are intersected and then evaluating the functions for each crater (block 10). Moon curvature compensation is made at this time also.

Slopes in two orthogonal directions are calculated by subtracting height values adjacent to a point and dividing by twice the range increment. Parallel slopes (block 11) refer to the direction parallel to the observer's line of sight and transverse slopes (block 12) to those perpendicular to the line of sight. The slopes and heights are used to find the outward normal to the surface. This in turn defines the angle of incidence of the sun's rays. Lambert's Law of Illumination is then applied (block 13), assuming perfectly diffuse reflection*. The sun calculations are performed moving outwardly from the sun along one of the previously generated rows or columns of data, thus causing the cardinal point restriction on the sun's position.

The intensity returns for each point on the row are projected to the plane of the observer (block 14). This calculation is done moving from far range in towards the observer and automatically accounts for the occlusion problem. Also, at this time the projected intensities are mapped into appropriate integer intensity codes for the SC 4020.

The integer values are transmitted to an Assembly Language subroutine (SHOLUN, block 15) which generates the SC 4020 instructions by bit shifting, and outputs a block of them each time it is called. An algorithm similar to the one for the "leading" edge works on the last two rows after the main loop has finished (block 18). The subroutine ENDPIC empties the output buffer and closes the output tape (block 19).

Table 1 is a legend for the charts and exhibits which follow. Each exhibit is a series of four related pictures generated by LUNSIM. Columns 2-9 and 14 of Table 1 are the run parameters for each picture. The "viewed area limits" in column 10-13 are calculated by LUNSIM and are printed as supplementary output. Charts A, B, and C correspond to Exhibits A, B, and C, respectively. Each subchart shows the relationship of the viewed area to the overall working area (the large grid) and the position of the observer and the sun for each picture.

---

* Obviously other functions of incidence angle could be used, thus accounting for different albedo from lunar features of differing texture.

Exhibit A is a simulated landing sequence with the observer "zeroing in" on a predetermined spot on the surface. The first of the series, A1, is a high-altitude view which encompasses about half the working area (nearly 1,000 sq km). Exhibits A1 and A2 were programmed to produce data with higher resolution than the other exhibits. The landing vehicle moves closer to the surface and the viewing angle is changed until, in Exhibit A4, it is looking straight down at a crater which is extremely close to the landing site.

Exhibit B represents a roving sequence. Actually the observer is at very low altitude and is looking ahead at a $35^{\circ}$ angle of elevation. As his ground coordinates are incremented, various surface features can be seen to pass beneath him. Chart B shows the movement relative to the entire map.

In the previous two exhibits the sun was arbitrarily positioned to the left side of the observer. Exhibit C demonstrates the effects of different relative observer and sun locations. In this exhibit approximately the same area is viewed in each picture. Notice the significant change in shadow when the sun is moved from a position perpendicular to the observer to a position opposite the observer.

The change in picture appearance arises from changing only the sun parameters: the crater map used by the computer is identical for all twelve pictures. The reader can realize the power and flexibility of this type of digital technique in visual simulation.

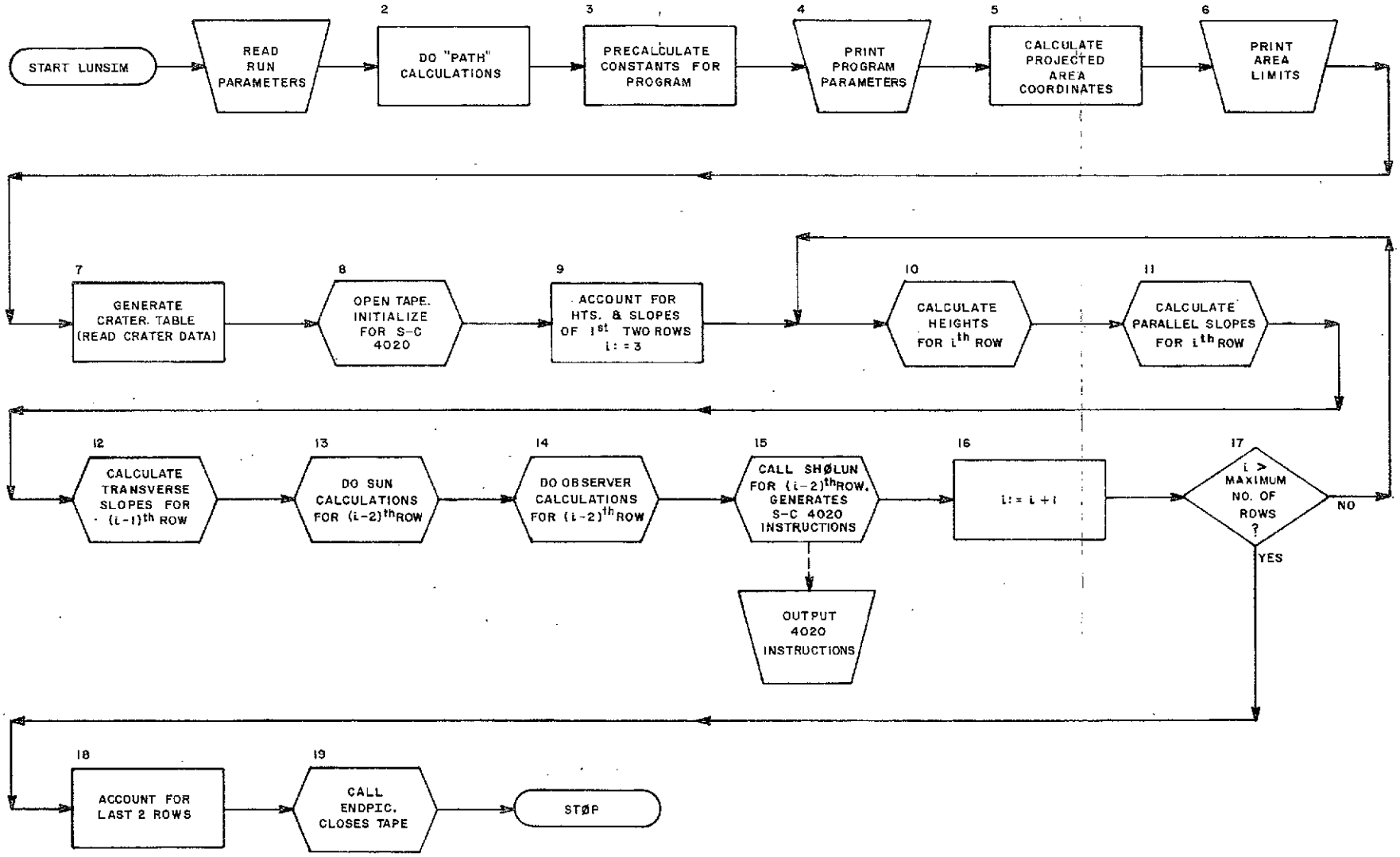



FIG. 1 - LUNSIM MACRO-FLOWCHART

FOLDOUT FRAME A

FOLDOUT FRAME B

TABLE 1 LEGEND FOR EXHIBITS

| Exhibit Number | OBSERVER PARAMETERS | | | | | | SUN PARAMETERS | | VIEWED AREA LIMITS, METERS (relative to SW corner of map) | | | | Number of grid points along one edge | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Azimuthal position (North, South, East, West) | Ground Coordinates, meters (relative to SW corner of map) | Altitude, meters | Elevation angle, degrees | Vertical field of view angle, degrees | Horizontal field of view angle, degrees | Azimuthal position (North, South, East, West) | Elevation angle, degrees | X - lower coordinate | Y - lower coordinate | X - maximum value | Y - maximum value | | |
| A1 | W | (-22,000, 25,900) | 35,000 | 43 | 30 | 30 | N | 20 | -129 | 16,522 | 43,825 | 35,278 | 512 | Landing sequence; touch down at $24^{\circ}5'E$, $1^{\circ}3'N$. Elevation angle changes until view is straight down at low altitude. |
| A2 | W | (11,000, 25,900) | 10,000 | 45 | 30 | 30 | N | 20 | 16,774 | 23,220 | 28,320 | 28,580 | 512 | |
| A3 | W | (19,368, 25,900) | 3,000 | 60 | 30 | 30 | N | 20 | 20,172 | 25,096 | 22,368 | 26,704 | 256 | |
| A4 | W | (21,100, 25,900) | 1,000 | 90 | 30 | 30 | N | 20 | 20,832 | 25,632 | 21,368 | 26,168 | 256 | |
| B1 | W | (21,100, 25,900) | 1,500 | 35 | 60 | 60 | N | 20 | 21,800 | 25,034 | 38,245 | 26,766 | 256 | Landing vehicle proceeds east at low altitude (simulates roving). Observer's position is incremented 4000 m in west-east direction. |
| B2 | W | (25,100, 25,900) | 1,500 | 35 | 60 | 60 | N | 20 | 25,800 | 25,034 | 42,245 | 26,766 | 256 | |
| B3 | W | (29,100, 25,900) | 1,500 | 35 | 60 | 60 | N | 20 | 29,800 | 25,034 | 46,245 | 26,766 | 256 | |
| B4 | W | (33,100, 25,900) | 1,500 | 35 | 60 | 60 | N | 20 | 33,800 | 25,034 | 50,245 | 26,766 | 256 | |
| C1 | S | (15,400, 12,300) | 20,000 | 70 | 60 | 60 | N | 15 | 3,853 | 8,773 | 26,947 | 36,135 | 256 | Effects of varying sun and observer positions relative to one area and to each other. |
| C2 | E | (24,600, 27,675) | 20,000 | 70 | 60 | 60 | N | 15 | 765 | 16,128 | 28,127 | 39,222 | 256 | |
| C3 | S | (15,400, 12,300) | 20,000 | 70 | 60 | 60 | W | 15 | 3,853 | 8,773 | 26,947 | 36,135 | 256 | |
| C4 | E | (24,600, 27,675) | 20,000 | 70 | 60 | 60 | W | 15 | 765 | 16,128 | 28,127 | 39,222 | 256 | |

FOLDOUT FRAME A

FOLDOUT FRAME B

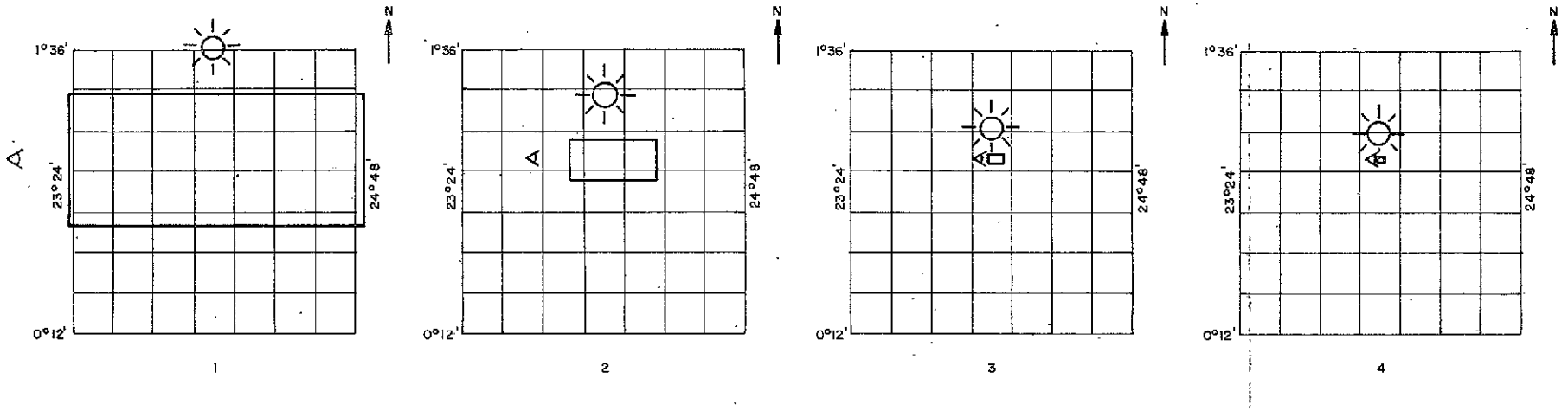



CHART A (CORRESPONDS TO EXHIBIT A) - LANDING SEQUENCE SHOWN IN EXHIBIT A RELATIVE TO ENTIRE WORKING AREA. ◁ DESIGNATES OBSERVER'S POSITION AND VIEWING DIRECTION; ☼ SHOWS SOLAR POSITION IN RELATION TO VIEWED AREA (HEAVY OUTLINED RECTANGLE). CRATER DATA GATHERED ONLY FOR AREA WITHIN MAP AREA SHOWN. FOR A MORE DETAILED DESCRIPTION OF THE RUN PARAMETERS WHICH THESE CHARTS REPRESENT SEE TABLE

FOLDOUT FRAME A

FOLDOUT FRAME B

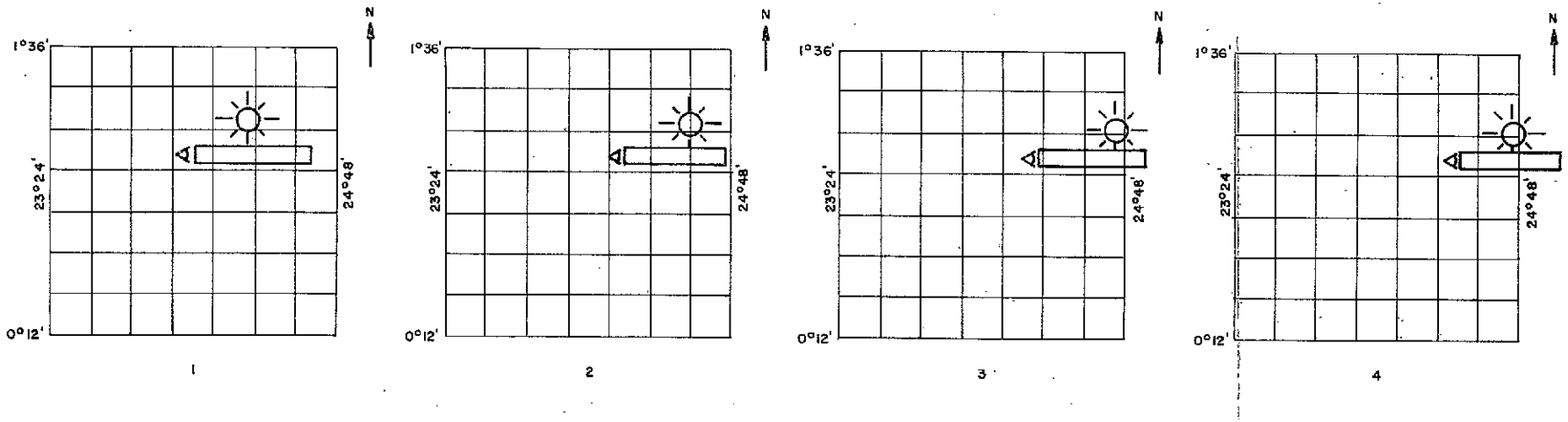



CHART B - SIMULATED ROVING SEQUENCE AS SHOWN IN EXHIBIT B. VEHICLE ACTUALLY IS FLYING AT LOW ALTITUDE.

FOLDOUT FRAME

A

FOLDOUT FRAME

B

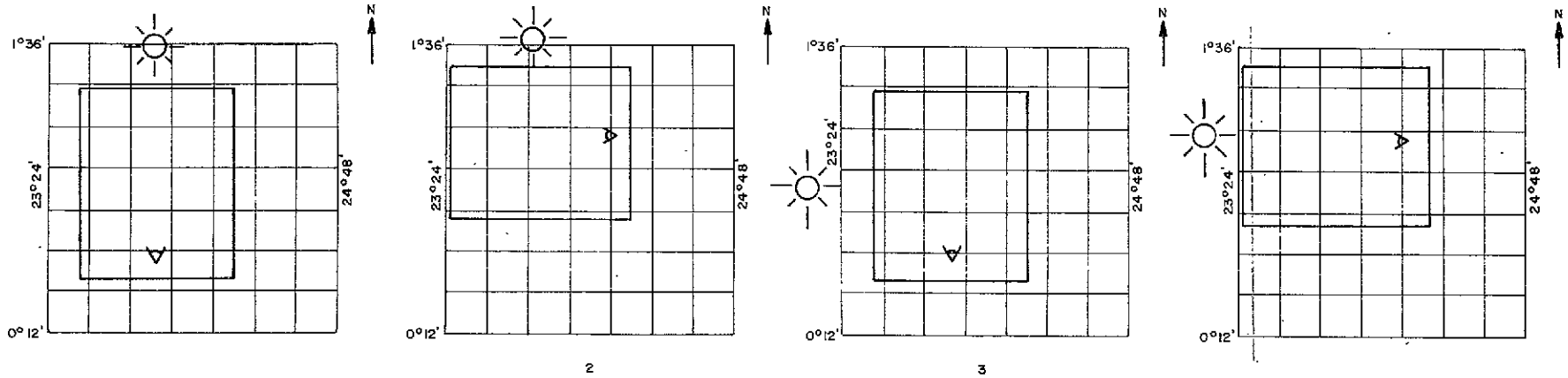



CHART C – OBSERVER, SUN, AND VIEWED AREA AS SEEN IN EXHIBIT C. DEMONSTRATES EFFECTS OF VARYING OBSERVER AND SUN POSITIONS WITH RESPECT TO A SPECIFIC PORTION OF THE LUNAR SURFACE.

FOLDOUT FRAME

A

FOLDOUT FRAME

B